usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

ACon^2: Adaptive Conformal Consensus for Provable Blockchain Oracles

Sangdon Park, *Georgia Institute of Technology;*
Osbert Bastani, *University of Pennsylvania;*
Taesoo Kim, *Georgia Institute of Technology*

https://www.usenix.org/conference/usenixsecurity23/presentation/park

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 32nd USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

USENIX'23 Artifact Appendix
ACon^2: Adaptive Conformal Consensus for Provable Blockchain Oracles

Sangdon Park[†] *Osbert Bastani*[*] *Taesoo Kim*[†]

[†]Georgia Institute of Technology [*]University of Pennsylvania

A Artifact Appendix

A.1 Abstract

Our paper proposes an online learning algorithm, called Adaptive Conformal Consensus. Our artifact consists of source code, datasets, docker files, and scripts to generate paper results. We aim for *Artifacts Available*, *Artifacts Functional*, and *Results Reproduced* badges.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

Code of our artifact will run a proposed machine learning algorithm over Python without external communication and a local blockchain with a forked Ethereum mainnet, so we do not expect to see any security, privacy, or ethical concerns. Note that in forking Ethereum mainnet, a script will use an author's API key for Alchemy, so we would not expect related security, privacy, and ethical issues.

A.2.2 How to access

Our artifacts are accessible via Github `https://github.com/sslab-gatech/ACon2/tree/AEStableVersion`[1].

A.2.3 Hardware dependencies

We expect a standard computing environment, i.e., a computing machine with CPU, HDD, and Internet access. In particular, a 4 or 5 core CPU machine would be preferred for multi-processing. The results and docker require about 4 GB HDD. Internet access is required to fork the Ethereum mainnet during experinents.

A.2.4 Software dependencies

Docker is required, as we provide docker images for reproducing our results.

A.2.5 Benchmarks

We include required datasets (i.e., USD/ETH data and INV/ETH data) into docker images; thus, additional actions to get datasets are not required.

A.3 Set-up

A.3.1 Installation

Our code repository is cloned via `git clone -depth 1 -branch AEStableVersion git@github.com:sslab-gatech/ACon2.git`. We provide docker files, so Docker needs to be installed. Other than these, all executions are done over docker images.

A.3.2 Basic Test

Once two docker images are installed and the code repository is cloned, (1) change the working directory to `python` and execute `./docker_scripts/docker_plot_INV_ETH_precomp.sh`; and (2) change the working directory to `solidity` and execute `./docker_scripts/plot_sim_precomp.sh`. These two scripts sould not introduce errors if set-up is right.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): *ACon^2 generates consensus sets that follows well USD/ETH price data change when $K = 1$. This is proven by the experiment (E1) whose results are illustrated in Figure 4(a).*

(C2): *ACon^2 generates consensus sets that follows well USD/ETH price data change when $K = 2$. This is proven by the experiment (E2) whose results are illustrated in Figure 4(b).*

(C3): *ACon^2 generates consensus sets that follows well USD/ETH price data change when $K = 3$. This is proven by the experiment (E3) whose results are illustrated in Figure 4(c).*

[1] git clone –depth 1 –branch AEStableVersion git@github.com:sslab-gatech/ACon2.git

(C4): *ACon^2 generates consensus sets that satisfy a desired pseudo-miscoverage rate over USD/ETH price data when $K = 1$. This is proven by the experiment (E4) whose results are illustrated in Figure 5(a).*

(C5): *ACon^2 generates consensus sets that satisfy a desired pseudo-miscoverage rate over USD/ETH price data when $K = 2$. This is proven by the experiment (E5) whose results are illustrated in Figure 5(b).*

(C6): *ACon^2 generates consensus sets that satisfy a desired pseudo-miscoverage rate over USD/ETH price data when $K = 3$. This is proven by the experiment (E6) whose results are illustrated in Figure 5(c).*

(C7): *ACon^2 generates reasonable small consensus sets over USD/ETH price data when $K = 3$. This is proven by the experiment (E7) whose results are illustrated in Figure 6(a).*

(C8): *a baseline algorithm σ-ACon^2 generates large consensus sets and conservative pseudo-miscoverage rates over USD/ETH price data when $K = 3$. This is proven by the experiment (E8) whose results are illustrated in Figure 9(a) and 9(b).*

(C9): *ACon^2 generates meaningful consensus sets under price manipulation, while trigger alarms for downstream applications over INV/ETH price data. This is proven by the experiment (E9) whose results are illustrated in Table 1 and Figure 1.*

(C10): *ACon^2 generates consensus sets that follows well INV/ETH price data change when $K = 1$. This is proven by the experiment (E10) whose results are illustrated in Figure 7(a).*

(C11): *ACon^2 generates consensus sets that follows well INV/ETH price data change when $K = 2$. This is proven by the experiment (E11) whose results are illustrated in Figure 7(b).*

(C12): *ACon^2 generates consensus sets that follows well INV/ETH price data change when $K = 3$. This is proven by the experiment (E12) whose results are illustrated in Figure 7(c).*

(C13): *ACon^2 generates consensus sets that satisfy a desired pseudo-miscoverage rate over INV/ETH price data when $K = 1$. This is proven by the experiment (E13) whose results are illustrated in Figure 8(a).*

(C14): *ACon^2 generates consensus sets that satisfy a desired pseudo-miscoverage rate over INV/ETH price data when $K = 2$. This is proven by the experiment (E14) whose results are illustrated in Figure 8(b).*

(C15): *ACon^2 generates consensus sets that satisfy a desired pseudo-miscoverage rate over INV/ETH price data when $K = 3$. This is proven by the experiment (E15) whose results are illustrated in Figure 8(c).*

(C16): *ACon^2 generates reasonable small consensus sets over INV/ETH price data when $K = 3$. This is proven by the experiment (E16) whose results are illustrated in Figure 6(b).*

(C17): *ACon^2 generates reasonable small consensus sets and achieves a desired pseud-miscoverage rate over local Ethereum network data when $K = 3$. This is proven by the experiment (E17) whose results are illustrated in Figure 10(a) and 10(b).*

(C18): *ACon^2 achieves a desired pseudo-miscoverage rate over local Ethereum network data with different K and α. This is proven by the experiment (E18) whose results are illustrated in Figure 11(a), 11(b), and 11(c).*

(C19): *ACon^2 uses a reasonable gas amount for computation. This is proven by the experiment (E19) whose results are illustrated in Table 2.*

A.4.2 Experiments

This section includes detailed instructions to reproduce results. Also, see `https://github.com/sslab-gatech/ACon2/tree/AEStableVersion`, which contains instructions with pre-computed data, which do not require heavy computation. Note that the measured compute-hours are estimated based on a server-level environment (i.e., 128 2GHz-CPUs with 500G memory); we expect one CPU with at least 500MB memory as minimal requirements, but the actual computation time could vary, depending on a HW setup.

Common preparation step.

1. Install Docker
2. Pull docker images via `dockerpullghcr.io/sslab-gatech/acon2:latest` and `dockerpullghcr.io/sslab-gatech/acon2-sol:latest`
3. Clone our code repository

(E1-8): *[0 human-minutes + 30 compute-hour + 5GB disk]: This experiment generates results for Figure 4, Figure 5, Figure 6(a), and Figure 9.*
How to: *First collect required data by executing a script.*
Preparation: *change the working directory to* `python`
Execution: *Run* `./docker_scripts/docker_run_USD_ETH.sh` *and Run* `./docker_scripts/docker_plot_USD_ETH.sh`
Results: *Ways to interpret results are described in* ***(E1-8)***

(E1): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 4(a).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 4(a), see* `output_docker/one_source_USD_ETH_UniswapV2_K_1_beta_0/figs/plot_ps.pdf`

(E2): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 4(b).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`

Results: *For Figure 4(b), see* `output_docker/two_sources_USD_ETH_UniswapV2_coinbase_K_2_beta_1/figs/plot_ps.pdf`

(E3): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 4(c).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 4(c), see* `output_docker/three_sources_USD_ETH_UniswapV2_coinbase_binance_K_3_beta_1/figs/plot_ps.pdf`

(E4): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 5(a).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 5(a), see* `output_docker/one_source_USD_ETH_UniswapV2_K_1_beta_0/figs/plot_miscoverage.pdf`

(E5): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 5(b).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 5(b), see* `output_docker/two_sources_USD_ETH_UniswapV2_coinbase_K_2_beta_1/figs/plot_miscoverage.pdf`

(E6): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 5(c).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 5(c), see* `output_docker/three_sources_USD_ETH_UniswapV2_coinbase_binance_K_3_beta_1/figs/plot_miscoverage.pdf`

(E7): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 6(a).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 6(a), see* `output_docker/one_source_USD_ETH_UniswapV2_K_1_beta_0_two_sources_USD_ETH_UniswapV2_coinbase_K_2_beta_1_three_sources_USD_ETH_UniswapV2_coinbase_binance_K_3_beta_1/figs/plot_size.pdf`

(E8): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 9(a,b).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 9(a), see* `output_docker/three_sources_OneSigma_USD_ETH_UniswapV2_coinbase_binance_K_3_beta_1/figs/plot_ps.pdf` *and for Figure 9(b), see* `output_docker/three_sources_OneSigma_USD_ETH_UniswapV2_coinbase_binance_K_3_beta_1/figs/plot_miscoverage.pdf`

(E9-16): *[0 human-minutes + 2 compute-hour + 5GB disk]: This experiment generates results for Table1, Figure 1, Figure 7, Figure 8, and Figure 6(a).*
How to: *First collect required data by executing a script.*
Preparation: *change the working directory to* `python`
Execution: *Run* `./docker_scripts/docker_run_INV_ETH.sh` *and Run* `./docker_scripts/docker_plot_INV_ETH.sh`
Results: *Ways to interpret results are described in **(E9-16)***

(E9): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Table 1 and Figure 1.*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Table 1, see* `stdout` *of* `./docker_scripts/docker_plot_INV_ETH.sh` *and for Figure 1, see* `output_docker/highlight/figs/plot_ps.pdf`

(E10): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 7(a).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 7(a), see* `output_docker/one_source_INV_ETH_SushiSwap_K_1_beta_0/figs/plot_ps.pdf`

(E11): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 7(b).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 7(b), see* `output_docker/two_sources_INV_ETH_SushiSwap_UniswapV2_K_2_beta_1/figs/plot_ps.pdf`

(E12): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 7(c).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 7(c), see* `output_docker/three_sources_INV_ETH_SushiSwap_UniswapV2_coinbase_K_3_beta_1/figs/plot_ps.pdf`

(E13): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 8(a).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 8(a), see* `output_docker/one_source_INV_ETH_SushiSwap_K_1_beta_0/figs/plot_miscoverage.pdf`

(E14): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 8(b).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 8(b), see* `output_docker/two_sources_INV_ETH_SushiSwap_UniswapV2_K_2_beta_1/figs/plot_miscoverage.pdf`

(E15): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 8(c).*
How to: *Check a generated figure.*

Preparation: *change the working directory to* `python`
Results: *For Figure 8(c), see* `output_docker/three_sources_INV_ETH_SushiSwap_UniswapV2_coinbase_K_3_beta_1/figs/plot_miscoverage.pdf`

(E16): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 6(b).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `python`
Results: *For Figure 6(b), see* `output_docker/one_source_INV_ETH_SushiSwap_K_1_beta_0_two_sources_INV_ETH_SushiSwap_UniswapV2_K_2_beta_1_three_sources_INV_ETH_SushiSwap_UniswapV2_coinbase_K_3_beta_1/figs/plot_size.pdf`

(E17-19): *[0 human-minutes + 30 compute-hour + 5GB disk]: This experiment generates results for Table 2, Figure 10, and Figure 11.*
How to: *First collect required data by executing a script.*
Preparation: *change the working directory to* `solidity`
Execution: *Enter into the docker image via* `./docker_scripts/enter.sh`*, execute* `./scripts/run.sh`*, execute* `./scripts/run_baseline.sh`*, exit from the docker image, and generate plots via* `./docker_scripts/plot_sim.sh`*.*
Results: *Ways to interpret results are described in* ***(E17-19)***

(E17): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 10(a,b).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `solidity`
Results: *For Figure 10(a), see* `output_docker/figs/acon2/plot-ps-K-3-alpha-0d01-iter-1.pdf` *and for Figure 10(b), see* `output_docker/figs/acon2/plot-error-var-K-3-alpha-0d01.pdf`

(E18): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Figure 11(a-c).*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `solidity`
Results: *For Figure 11(a), see* `output_docker/figs/acon2/plot-error-var-K-3-alphas.pdf`*, for Figure 11(b), see* `output_docker/figs/acon2/plot-error-var-K-4-alphas.pdf`*, and for Figure 11(c), see* `output_docker/figs/acon2/plot-error-var-K-5-alphas.pdf`*,*

(E19): *[1 human-minutes + 1 compute-minutes + 5GB disk]: This experiment generates results for Table 2.*
How to: *Check a generated figure.*
Preparation: *change the working directory to* `solidity`
Results: *For Table 2, see* `stdout` *of* `./docker_scripts/plot_sim.sh`*.*

In all of the above blocks, please provide indications about the expected outcome for each of the steps (given the suggested hardware/software configuration above).

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20220926. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at `https://secartifacts.github.io/usenixsec2023/`.