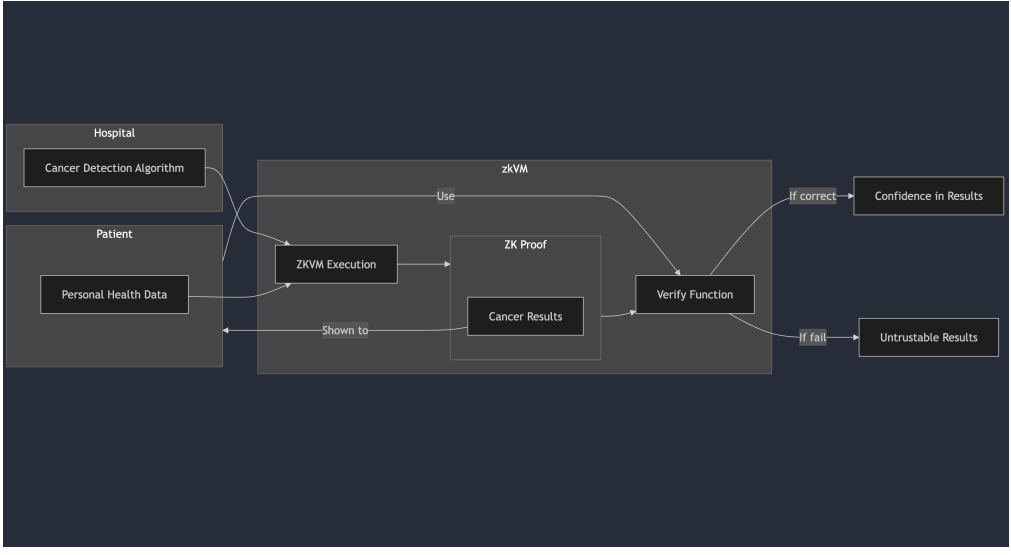
# Benchmarking ZK Virtual Machines for Privacy-Preserving Machine Learning Applications

Lawrence Lim, Siddhartha Tuladhar, Brandon Gao

NYU Shanghai

# Background


Hospital
Cancer Detection Algorithm
Patient
Personal Health Data
zkVM
Use
ZKVM Execution
ZK Proof
Cancer Results
Shown to
Verify Function
If correct
Confidence in Results
If fail
Untrustable Results

# Purpose

- zkVMs allow you to generate proofs for the execution of arbitrary instructions
- Identify: Which zkVM on the market is best equipped to handle computationally intensive workloads
- Highlight: Real-world ML application that leverages the succinctness property of zkVMs
- Compare: How underlying zkVM architecture impacts present / future performance, development experience

# Additional Details

- It's hard to verify the performance of AI models trained on private data
- Without the private test data, there is no way to verify the accuracy of the model
- However, with zkVMs, we can create a proof that attests to the models accuracy that is fast to verify
- There is a lack of up-to-date third party benchmarks on zkVM performance on real-world workloads

# Project Overview

- Python & Rust codebase, comparative analysis paper
- Train a regression model on real world transaction data
- Export model weights, test data and do inference on test data in zkVM
- Benchmark proof time, verification time on different architectures with different input sizes

# Data Source

- Retail Transaction Data: Kaggle dataset that collects retail transaction all across the US.
- Features: CustomerID, frequency, monetary, recency, Price DiscountApplied(%)
- Feature to predict: spend_90_days: spending in the next quarter

# Models

- Goal : Predicting the spending of the next quarter, based on previous quarter
- Linear Regression: explores the linear relationship between features and the target.

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

- Ridge Regression: A regularized version of linear regression that penalizes large coefficients to avoid overfitting

$$\text{Loss} = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

- Polynomial Ridge Regression: Explore non-linear relationships between features and the target, but with a regularization term that prevents overfitting by shrinking the model's coefficients
- R^2 Means Error: 0.8, Mean Square Error: 21

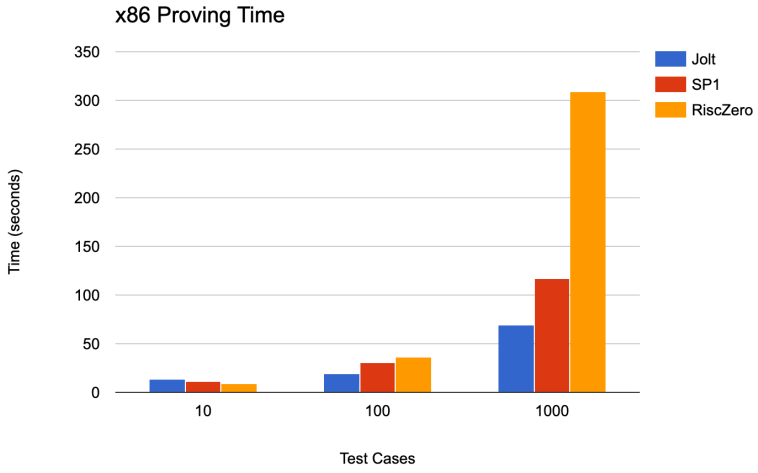
# Rust Implementation

- Implemented in Rust without the Standard Library (stdlib includes I/O, additional data structures)
- Manual implementation of Ridge Regression
- Cannot implement more complex models because using stdlib significantly increases proving

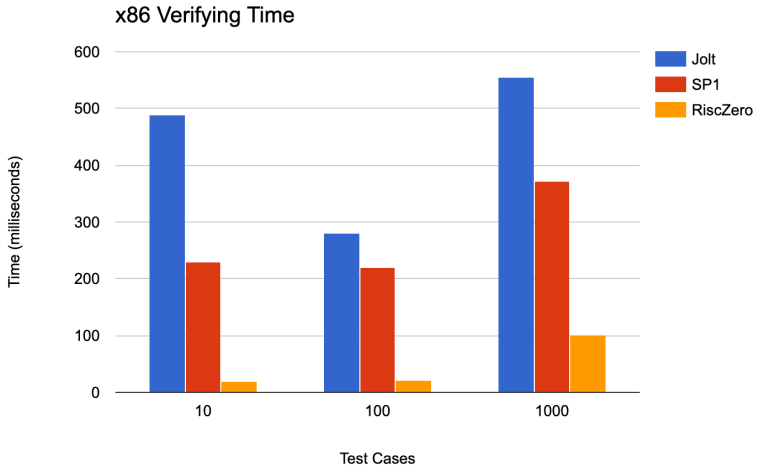
# Different ZKVMs

- Risc0: First production ready zkVM, optimized for use on GPUs, as well as a remote prover service
- SP1: Builds off the architecture of Risc0 with performance improvements through precompiles (manual optimizations)
- Jolt: Newest zkVM that runs precomputes a large lookup table to optimize performance

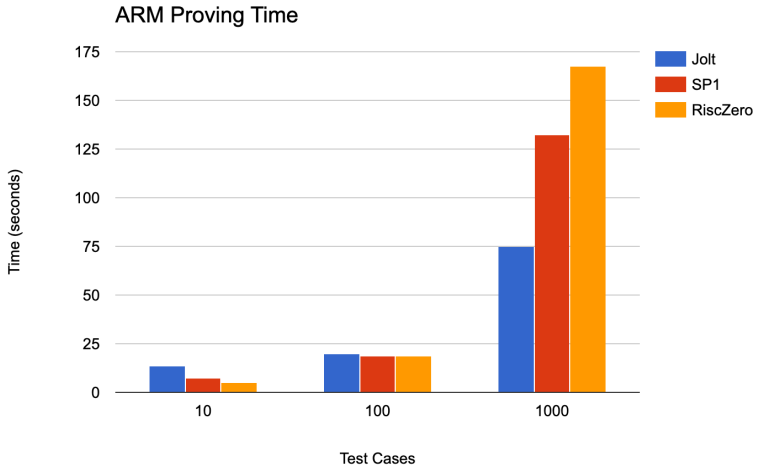
# x86 Proving Time



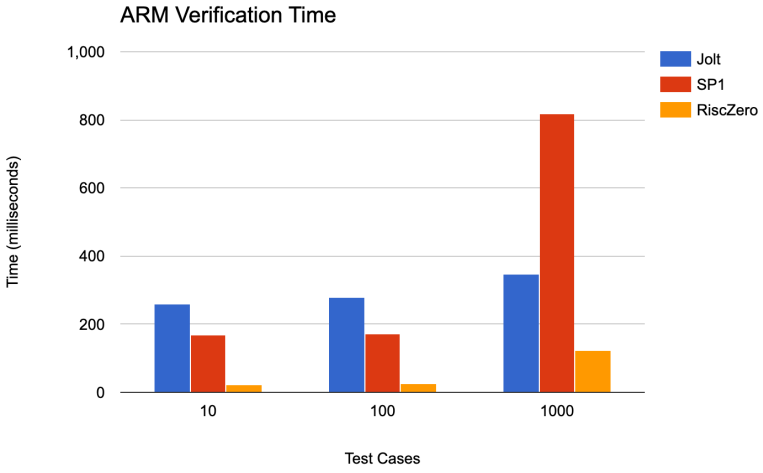
# x86 Verifying Time



# ARM Proving Time



# ARM Verification



# Next Steps

- Benchmarking without precompiles
- CPU vs. GPU performance
- Memory usage benchmarking
- Qualitative analysis