VeriRegime: On-Chain Verifiable Market Regime Classification with zkML

Lin Boyi 123090327

The Chinese University of Hong Kong, Shenzhen

DDA4220: Deep Learning

October 15, 2025

Abstract

Zero-knowledge machine learning (zkML) enables verifiable inference—allowing model predictions to be mathematically proven correct without exposing parameters or data. While most zkML research focuses on system-level efficiency and general AI benchmarks, few studies explore interpretable, domain-specific deployments. This project proposes **VeriRegime**, an on-chain verifiable deep learning framework for **market regime classification**—determining bullish, bearish, or sideways conditions in crypto markets. By combining compact neural architectures, quantization-aware training, and zero-knowledge proof systems (Groth16, Halo2), VeriRegime aims to deliver verifiable, low-latency inference deployable as a blockchain oracle. The work extends an earlier VeriRegime prototype (Lin, 2025) and investigates zkML-friendly model design, adaptive quantization, and activation optimization tailored to financial time-series data.

1 Introduction

Deep learning has achieved notable success in financial forecasting, yet models deployed in trading stacks often remain opaque and unverifiable. In decentralized finance (DeFi) and autonomous trading agents, this opacity introduces trust risk: counterparties must take model claims on faith, undermining transparency mandates and auditability. Zero-knowledge proofs (ZKPs), particularly succinct non-interactive arguments of knowledge (zkSNARKs), offer a cryptographic mechanism to verify computation correctness without revealing proprietary inputs or weights.

Integrating ZKPs with machine learning yields *zero-knowledge machine learning* (zkML), where every inference can be accompanied by a proof of correctness. Existing zkML pipelines demonstrate technical feasibility on canonical benchmarks, but practical, domain-specific financial applications remain underexplored. This project targets that gap by delivering **on-chain ver-**

ifiable regime classification—transforming model predictions of bullish, bearish, or sideways markets into proof-carrying statements that smart contracts can trust.

The proposed research focuses on designing zkML-friendly neural architectures, compiling them into efficient circuits, and deploying verifiers on-chain. Through end-to-end evaluation, VeriRegime will surface the trade-offs between accuracy, proof size, and latency that govern whether verifiable AI signals can support real-world DeFi strategies and governance.

2 Literature Review

System-level zkML. Early frameworks such as **zkCNN** (Lai et al., 2023) and **ZKML** (ZKML, 2024) introduced pipelines that convert neural networks into arithmetic circuits compatible with Groth16 proofs. **Artemis** (Artemis, 2024) refined commitment handling with commit-and-prove SNARKs, reducing verifier cost and proof size. More recently, **zkLLM** (Zhang et al., 2024) extended zkML support to transformer-based models using lookup-optimized attention constraints.

Proof-system innovation. Frameworks such as **Nova** (Nova, 2023) and **Plonky2** (Plonky2, 2023) explored recursion and folding schemes for faster incremental proof aggregation. Halo2's recursion-friendly design and lookup tables make it attractive for financial time-series inference, where regime states evolve continuously and require sliding-window evaluation.

Blockchain integration. Projects including the Halo2-based **EZKL** toolkit and the **Mina zkML Library** (Mina Protocol, 2025) provide ONNX-to-circuit compilers and code generation for verifier smart contracts. Industrial efforts such as **Modulus Labs** (Modulus Labs, 2025) demonstrate AI-on-chain services, showcasing the commercial momentum behind zkML. Nonetheless, literature still lacks domain-specific analysis for financial forecasting or risk management, leaving a research gap that VeriRegime aims to address.

3 Research Questions

1) How can compact neural architectures (e.g., MLP or lightweight Transformer) be efficiently transformed into zk-verifiable circuits for market regime classification?

2) What trade-offs emerge between model accuracy, proof size, and proving time under varying quantization levels and activation functions?

3) How does on-chain verification (EVM vs. Mina) influence cost, latency, and deployment complexity?

4) Which activation function redesigns or approximations best balance zk-friendliness and predictive power?

5) Can adaptive per-layer quantization strategies improve zkML inference efficiency without significant accuracy degradation?

4 Methods

4.1 Model Design

The base model will classify crypto market regimes (bullish, bearish, or sideways) using market and on-chain features: moving-average differentials (EMA5–EMA10), funding rate, open interest change, realized volatility, stablecoin flows, and address growth. A compact MLP (three hidden layers with ReLU activations) will be trained with quantization-aware methods and exported to ONNX format for zk compilation. Teacher-student distillation from an LSTM baseline will emphasize interpretability while retaining predictive strength.

4.2 zkML Compilation and Proof Generation

The ONNX model will be compiled via the **EZKL** framework (Halo2 backend). Nonlinearities such as ReLU will be approximated with polynomial or selection-gate constraints, and softmax will be replaced by argmax to avoid expensive divisions. For each inference, the circuit will generate:

- A proving key and verifying key pair (Groth16, Halo2),
- Commitments to model weights and inputs using Poseidon hashes,
- Public inputs: `model_commit`, `input_commit`, and `regime_output`.

4.3 Verification and Evaluation

Proofs will be validated both off-chain and on-chain. On-chain verifiers (EVM and Mina) will measure gas usage, latency, and scalability; off-chain verification will benchmark throughput on commodity hardware. Key performance metrics include accuracy, proof size, proving time, verification latency, and operational cost measured in gas or Mina fees.

4.4 Experimental Framework

Experiments will evaluate:

- Baseline architectures: LSTM teacher vs. zk-compatible MLP student,
- Quantization ablation: 4–16 bit, uniform vs. adaptive bit-width allocation,
- Activation study: ReLU, polynomial, sigmoid, and lookup-based variants,
- Proof-system comparison: Groth16, Halo2, and exploratory tests with Plonky2 recursion.

Results will be mapped onto an accuracy–efficiency Pareto frontier to highlight feasible deployment regimes.

4.5 Model Optimization for zkML Compatibility

To further improve verifiability, the project investigates zk-friendly optimization techniques:

a) **Quantization-Aware Training (QAT):** Simulate quantization noise during training to preserve accuracy under integer constraints.

b) **Polynomial Activation Networks:** Replace ReLU with low-degree polynomial approximations to reduce arithmetic gate counts.

c) **Pruning and Distillation:** Transfer structure from non-zk-compatible teachers (e.g., LSTM) to compact students with minimal redundant parameters.

d) **Adaptive Quantization:** Dynamically adjust per-layer bit-width to optimize proof time without significant loss in accuracy.

5 Timeline

Stage	**Task**	**Timeframe (2025)**
Week 7–8	Data curation, baseline training, QAT experiments	Oct
Week 8–9	Circuit compilation, proof generation, quantization ablation	Oct
Week 10–11	On-chain verifier deployment (EVM, Mina) and benchmarking	Nov
Week 12–13	Extended experiments (activation design, adaptive quantization)	Nov
Week 14	Final report, presentation, and optional workshop submission	Nov

Table 1: Project Schedule for Fall 2025

6 Significance of Research

This project advances the intersection of AI verifiability and decentralized finance by demonstrating a domain-specific zkML solution. Contributions include:

- The first **market-focused zkML application** for crypto regime detection,
- A systematic study of **zkML-friendly architectures**, activations, and quantization strategies,
- Empirical measurement of the **accuracy–proof cost Pareto frontier** for financial inference,
- Deployment of an **on-chain verifiable oracle** that integrates AI inference with blockchain trust assurances,
- Comparative analysis of Groth16, Halo2, and Plonky2 proof systems in a financial context.

Beyond demonstrating feasibility, VeriRegime establishes design principles for future **trustworthy, auditable AI systems** underpinning DeFi governance, DAO treasury management, and algorithmic risk controls.

References

Artemis Team. (2024). Commit-and-Prove SNARKs for Efficient zkML Verification. *CRYPTO 2024*.

Lai, R. et al. (2023). zkCNN: Efficient Verification of Convolutional Neural Networks. *USENIX Security 2023*.

Mina Foundation. (2025). Mina zkML Library Developer Guide. `https://minaprotocol.com/blog/minas-zkml-library-developer-guide`

Modulus Labs. (2025). Bringing AI On-chain: zkML in Production. *Medium*, April 2025.

Bünz, B. et al. (2023). Nova: Recursive Zero-Knowledge Proofs of Knowledge. *IACR ePrint 2023*.

Polygon Team. (2023). Plonky2: Recursive SNARKs for Fast Proof Composition. *Polygon Labs Whitepaper*.

ZKML Authors. (2024). ZKML: An Optimizing System for ML Inference in Zero-Knowledge Proofs. *EuroSys 2024*.

Zhang, W. et al. (2024). zkLLM: Verifiable Inference for Large Language Models. *arXiv preprint arXiv:2404.16109*.