RESOURCE SCHEDULING FOR REAL-TIME MACHINE LEARNING

Suyash Vardhan Singh, University of South Carolina, Columbia, South Carolina, USA ;
Iftakhar Ahmad, University of South Carolina, Columbia, South Carolina, USA;
Austin R.J. Downey, University of South Carolina, Columbia, South Carolina, USA ;
Miaoqing Huang, University of Arkansas, Fayetteville, Arkansas, USA;
David Andrews, University of Arkansas, Fayetteville, Arkansas, USA ;
Jason D. Bakos, University of South Carolina, Columbia, South Carolina, USA

U.S. National Science Foundation

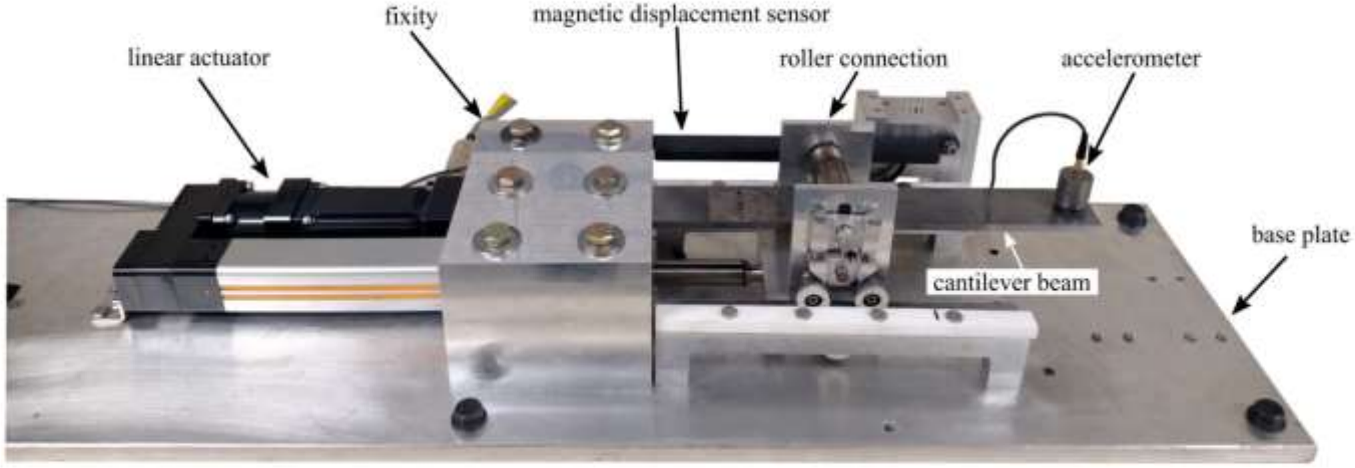
UNIVERSITY OF South Carolina
Molinaroli College of Engineering and Computing

UNIVERSITY OF ARKANSAS

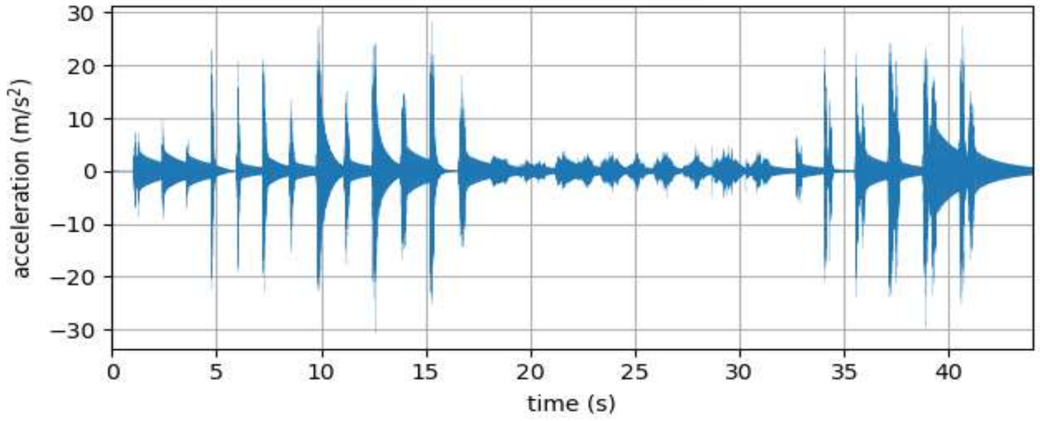
MOTIVATION & CHALLENGE

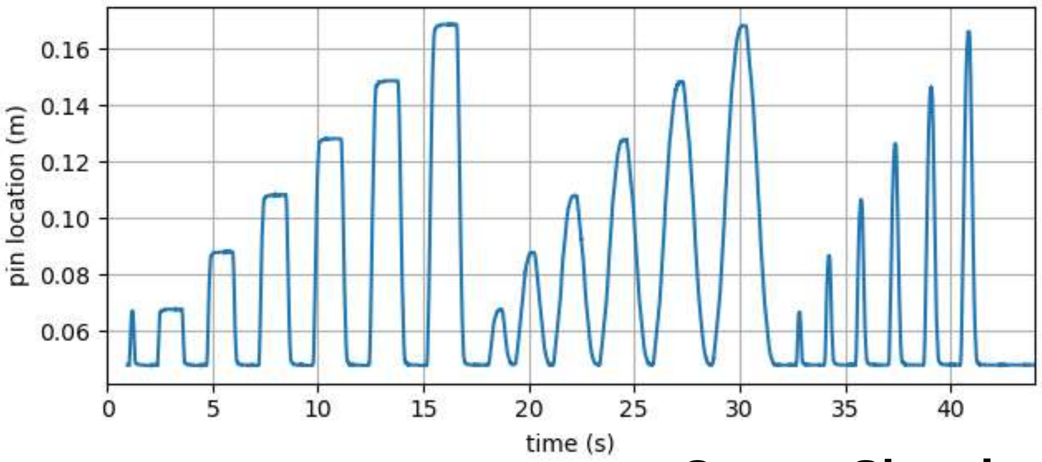
- **Challenge**: Deploying ML models in real-time high-rate cyber-physical systems.
    - Requires **sub-millisecond inference times**.
    - Operates under **stringent resource constraints** on FPGAs.
- **Deployment Process**:
    - Optimize **neural network configurations**.
    - Tune **hardware parameters** for:
        - **Resource efficiency** to fit FPGA limitations.
        - **Ultra-low latency** performance.

REAL-TIME HIGH-RATE DATASET - DROPBEAR



Input Signal





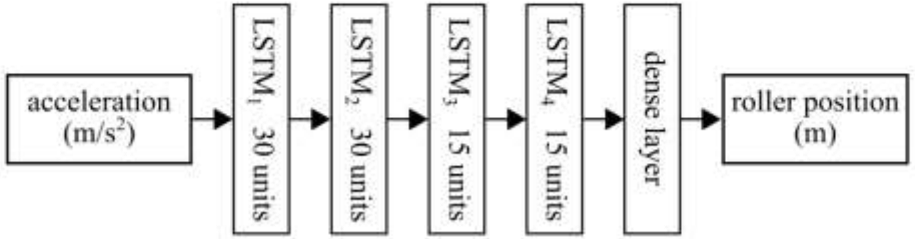
Output Signal

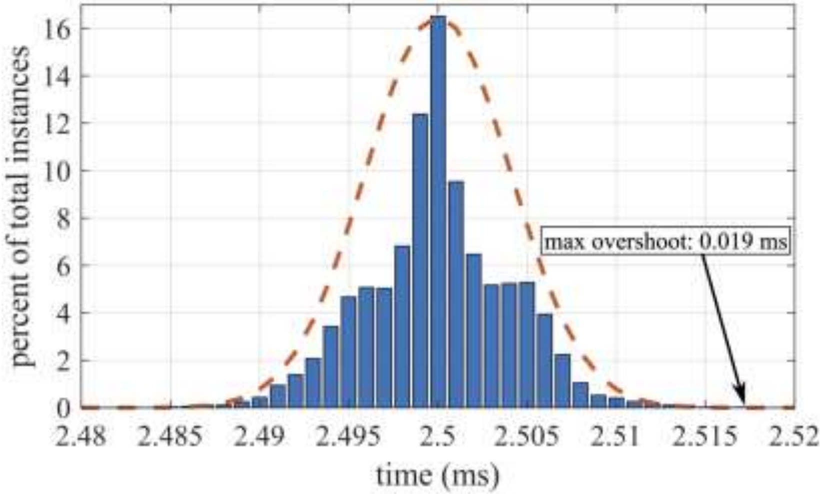
RELATED WORK ON DROPBEAR

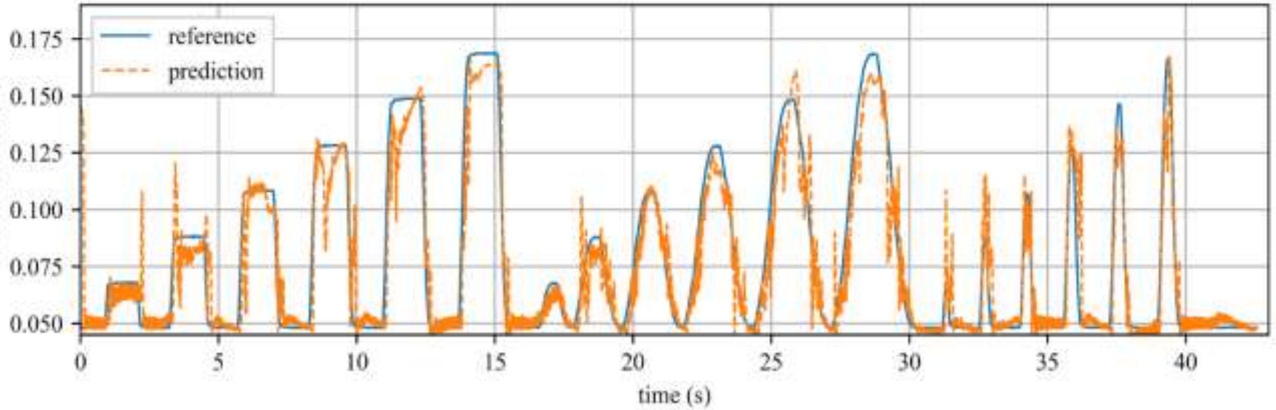
- **Accelerating LSTM-based High-Rate Dynamic System Models**
  - publication: 33rd International Conference on Field Programmable Logic and Applications (FPL 2023)
  - authors: Ehsan Kabir, Daniel Coble, Joud N. Satme, Austin R.J. Downey, Jason D. Bakos, David Andrews, Miaoqing Huang
- **Progress Towards Data-Driven High-Rate Structural State Estimation on Edge Computing Devices**
  - publication: In Volume 10 34th Conference on Mechanical Vibration and Sound (VIB). American Society of Mechanical Engineers, aug 2022. doi 10.1115/detc2022-90118.
  - authors: Joud Satme, Daniel Coble, Braden Priddy, Austin R.J. Downey, Jason D. Bakos, Gurcan Comert

RELATED WORK ON DROPBEAR (CONTD.)

- Rigid structure with single or multiple LSTM cells/layers and one output dense layer
- None deployed to FPGAs yet
- Networks had to be small to run in software on the low-power CPUs
  - Still performance wasn't good, with max sample rate at 400 Hz
- Same dataset for training and testing
  - Results may not be generalizable





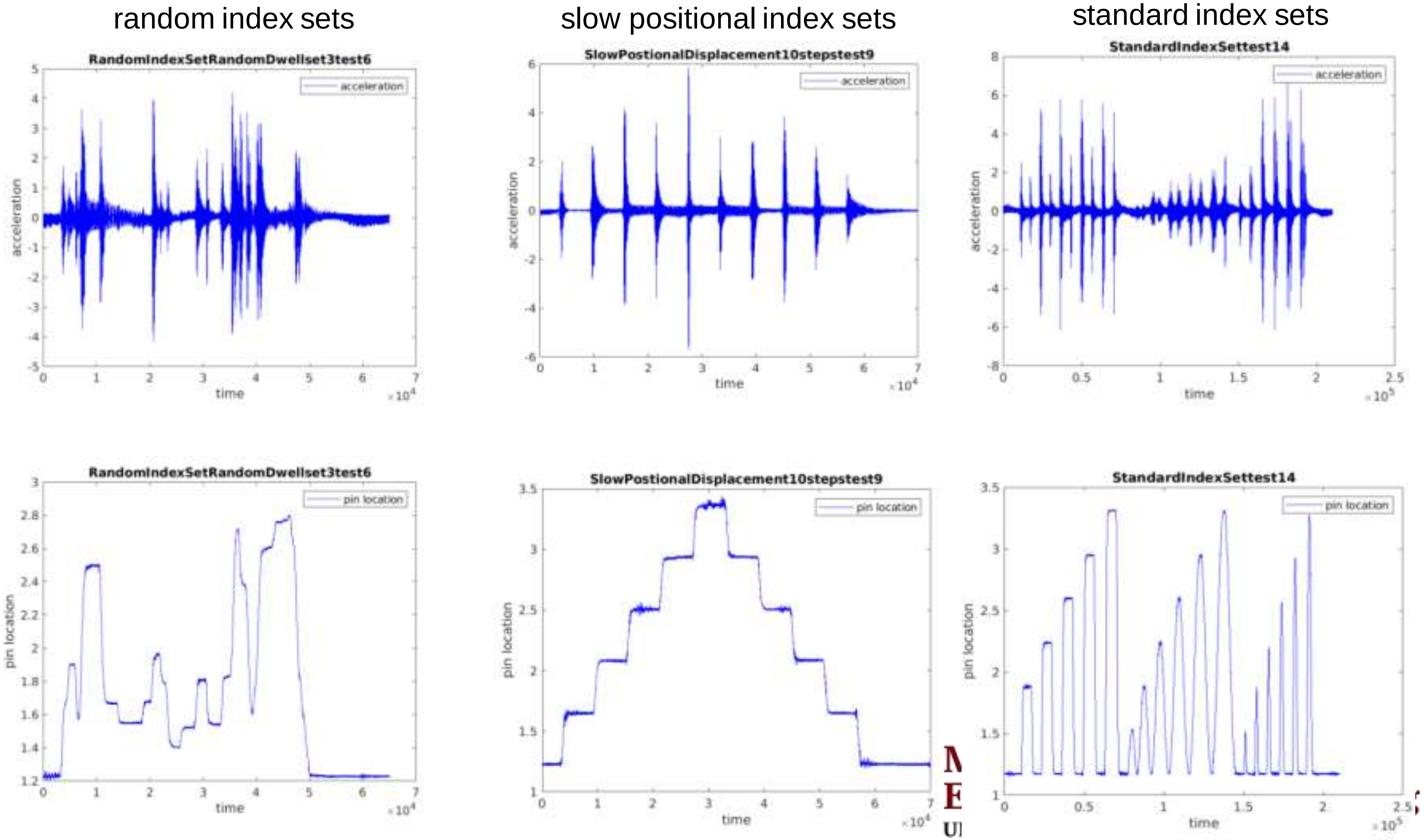


NEW DATASET - DATASET 8

- Sample rate: 5 KHz
- 3 Categories with 150 different experimental runs

Github Repo:

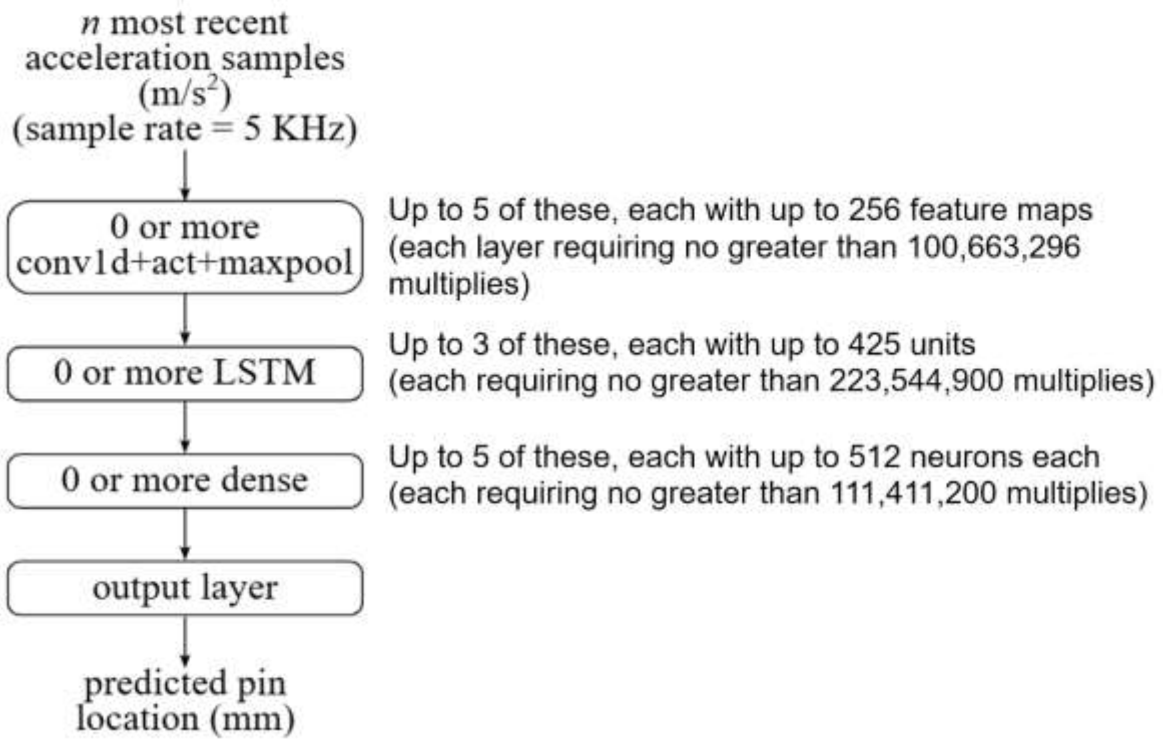
https://github.com/High-Rate-SHM-Working-Group/Dataset-8-DROPBEAR-Acceleration-vs-Roller-Displacement



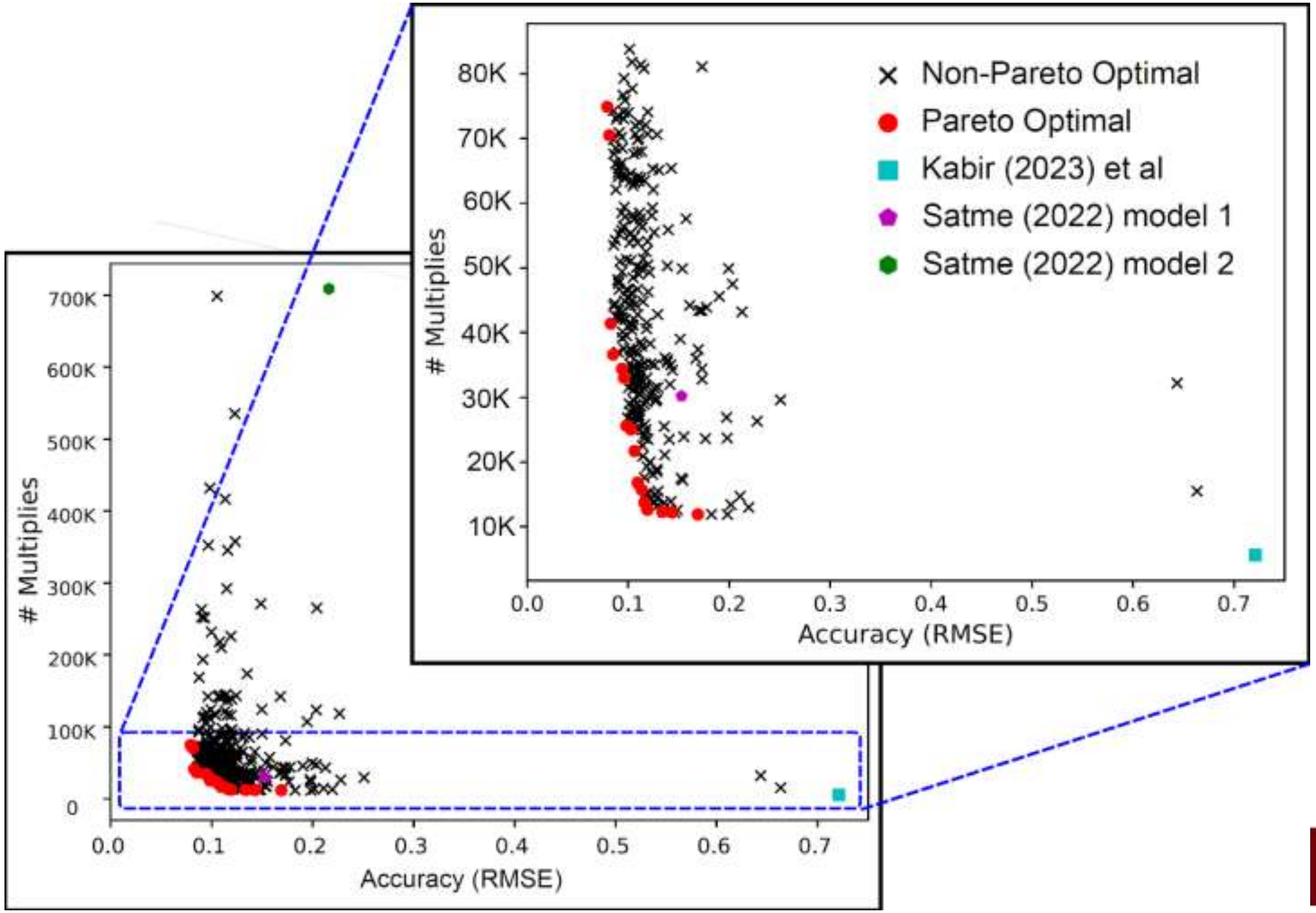
OUR APPROACH

- Random selection of 15 datasets from each of three categories
  - Training: 12
  - Testing 3
  - Shuffle the data
  - Split training data into a 70-30% ratio for training and validation
- Optuna framework
  - Hyperparamet optimization
  - Multi-objective Bayesian optimization
  - Objective function
    - RMSE and Workload (# multiplies)
  - Determine Pareto optimal set

Range of networks

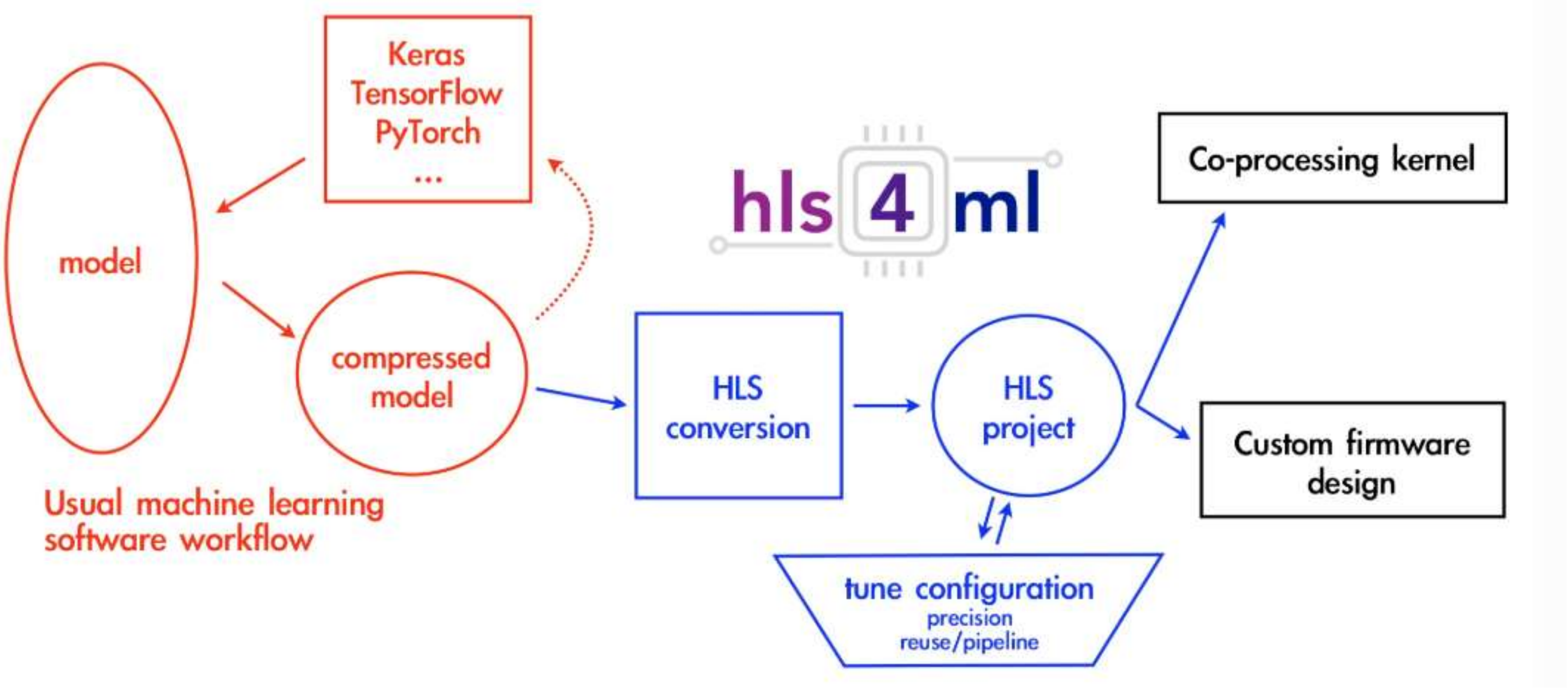


PARETO OPTIMAL NETWORKS



Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA

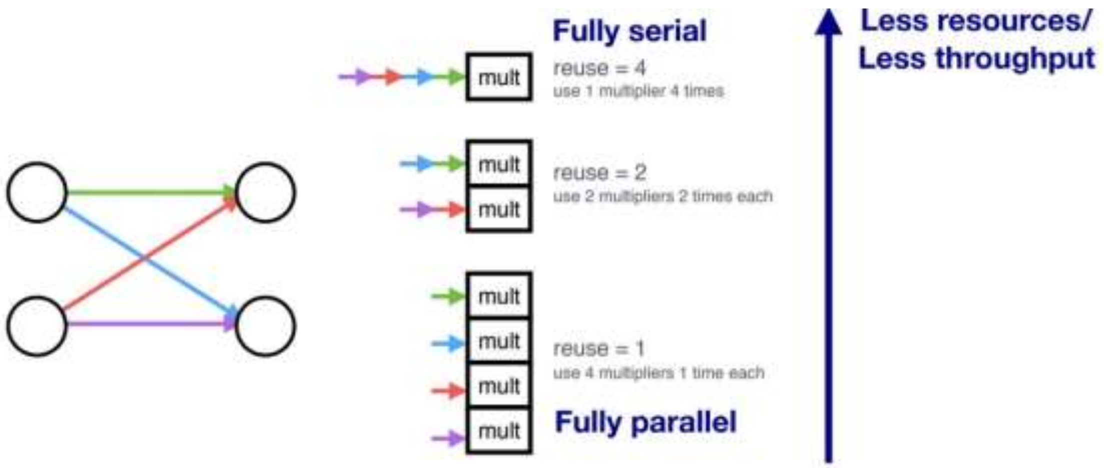
HLS4ML BACKGROUND



REUSE FACTOR

- Latency Strategy
- Resource Strategy

Latency strategy II=1
(ReuseFactor=1)




```
#pragma HLS PIPELINE II=CONFIG_T::reuse_factor

// #pragma HLS ARRAY_PARTITION variable=weights complete // remove this line for now, it breaks compression sometimes
#pragma HLS ARRAY_PARTITION variable=biases complete
#pragma HLS ARRAY_PARTITION variable=mult complete
#pragma HLS ARRAY_PARTITION variable=acc complete

#pragma HLS ALLOCATION operation instances=mul limit=CONFIG_T::multiplier_limit

o the matrix-multiply
uct1:
for (int ii = 0; ii < CONFIG_T::n_in; ii++) {
    cache = data[ii];
Product2:
    for (int jj = 0; jj < CONFIG_T::n_out; jj++) {
        int index = ii * CONFIG_T::n_out + jj;
        mult[index] = CONFIG_T::template product<data_T, typename CONFIG_T::weight_t>::product(cache, weights[index]);
    }
}
```


Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA

LATENCY STRATEGY : REUSE FACTOR

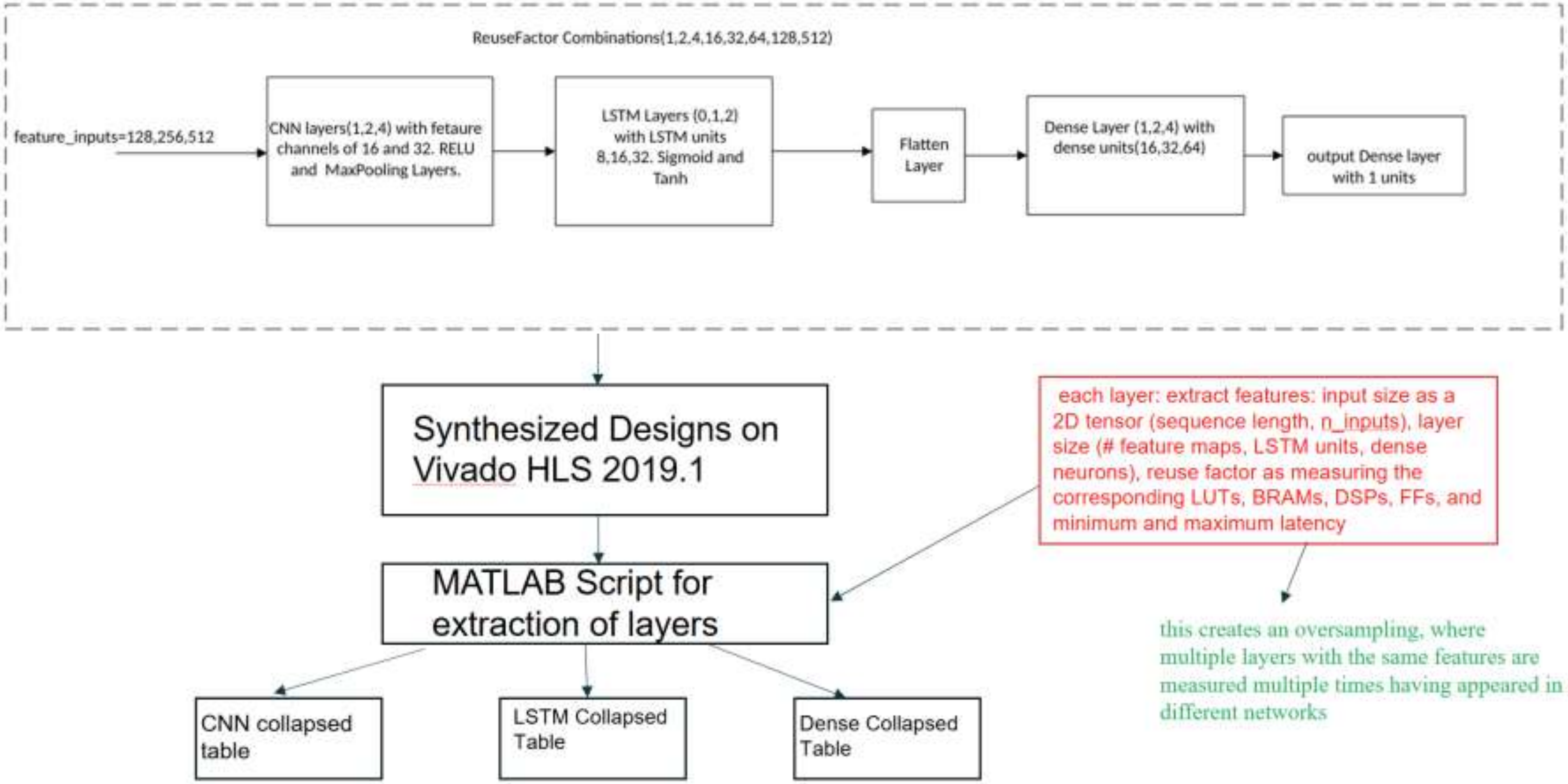
```
MultLoop:
    for (int im = 0; im < block_factor; im++) {
        #pragma HLS UNROLL

        acc[out_index] += static_cast<typename CONFIG_T::accum_t>(
            CONFIG_T::template product<data_T, typename CONFIG_T::weight_t>::product(data[in_index], weights[w_index]));

        // Increment w_index
        w_index += rufactor;
        // Increment in_index
        in_index += rufactor;
        if (in_index >= nin) {
            in_index = ir;
        }
        // Increment out_index
        if (acc_step + 1 >= multscale) {
            acc_step = 0;
            out_index++;
        } else {
            acc_step++;
        }
    }
}
```

Block factor = $\frac{[n\_in * n\_out]}{\text{Reuse Factor}}$

Unroll Factor

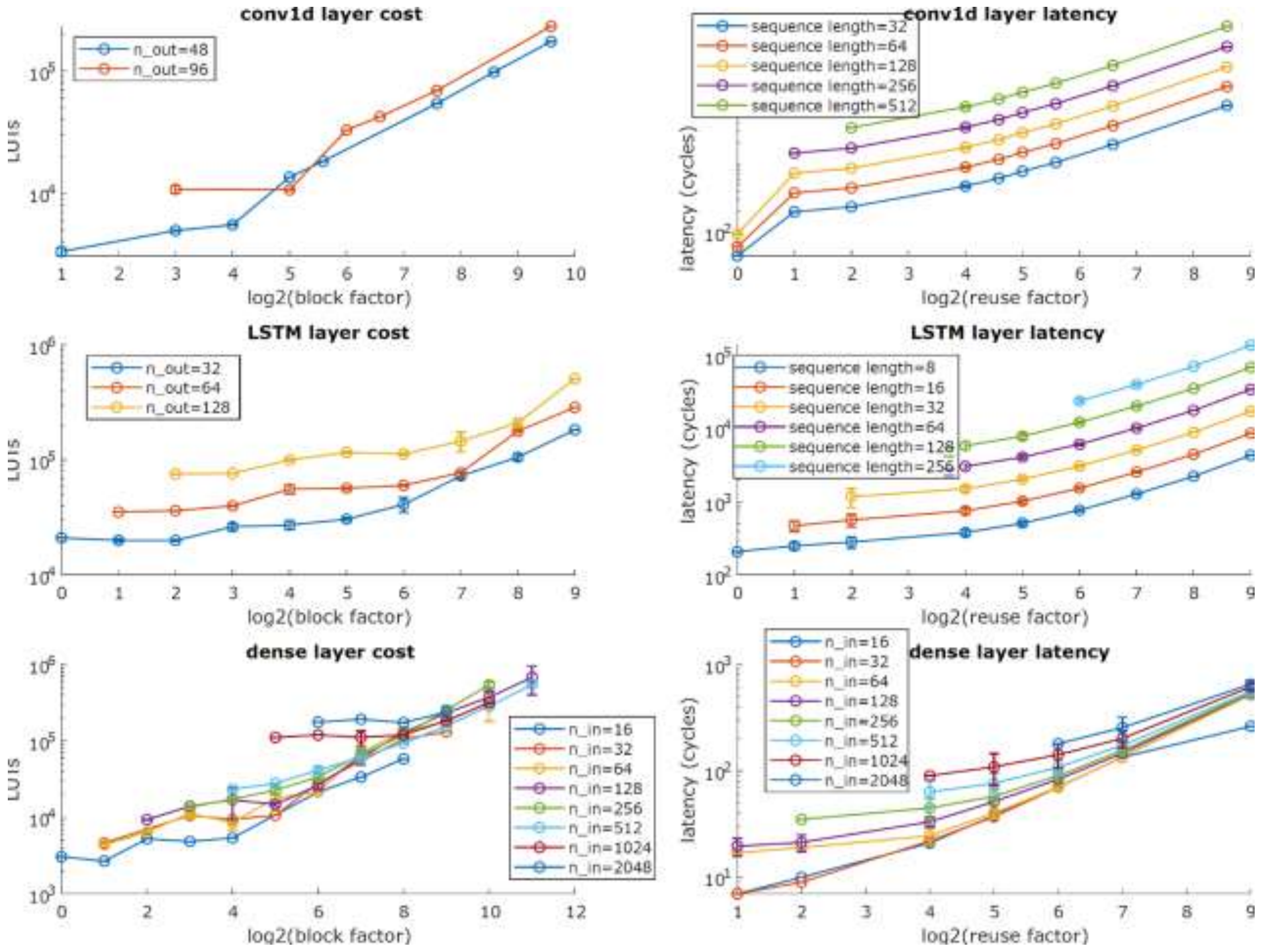
DATA GENERATION

Dataset:11000 synthesized networks



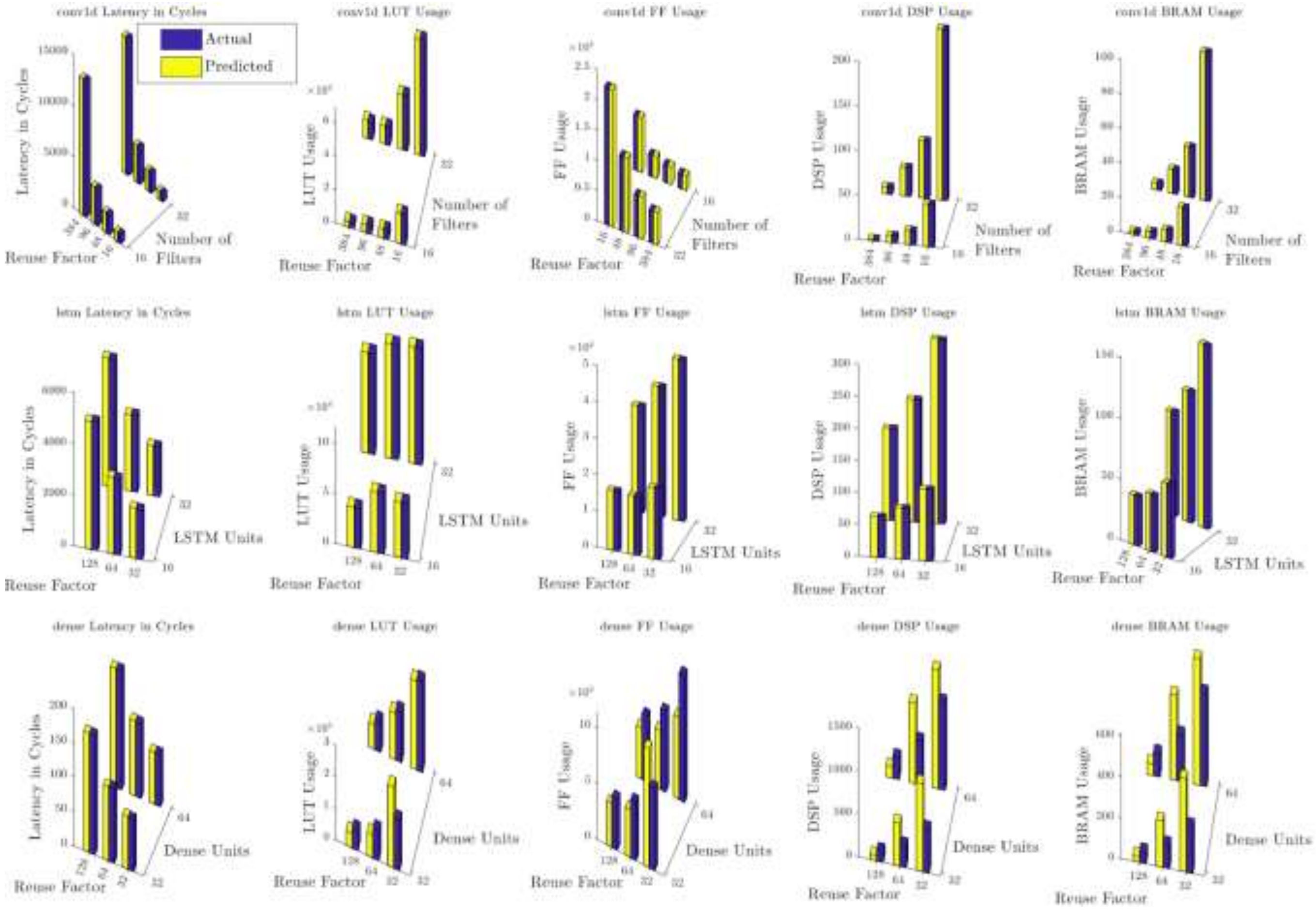
Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA

LUT COST AND LATENCY WHEN SCALING SIZE OF HARDWARE



Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA

PERFORMANCE AND COST MODEL RESULTS FOR VARIOUS LAYERS



Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA

PREDICTION METRICS ON TEST DATA

Layer	Metric	R^2 Score	MAPE	RMSE %	Value Range
Convolutional	BRAM	0.9976	0.44	6.76	0 - 342
	LUT	0.9988	2.35	3.95	2121.82 - 231963
	FF	0.9995	0.60	1.84	1042 - 75576
	DSP	0.9979	1.21	6.86	1 - 768
	Latency	0.9999	0.09	0.71	45 - 101910
LSTM	BRAM	0.9371	11.98	23.37	16 - 489
	LUT	0.9800	1.36	11.16	18580.714 - 286843
	FF	0.9826	1.23	10.06	7680.33 - 87131
	DSP	0.9780	1.65	15.54	26 - 1072
	Latency	0.9988	2.59	6.00	209 - 140545
Dense	BRAM	0.9954	0.13	11.48	0 - 910
	LUT	0.9921	0.14	15.17	1203 - 1079840
	FF	0.9989	0.09	4.89	1269 - 206076
	DSP	0.9956	0.12	13.54	1 - 2048
	Latency	0.9931	4.20	10.18	7 - 793

The R^2 score, also called the **coefficient of determination**, measures how well a regression model explains the variability of the dependent variable (target).

- **Formula**:
$$R^2 = 1 - \frac{\text{Sum of Squared Residuals (SSR)}}{\text{Total Sum of Squares (TSS)}}$$
  - **SSR**: The sum of squared differences between predicted and actual values.
  - **TSS**: The sum of squared differences between actual values and their mean.
- **Key Points**:
  - A higher R^2 indicates better predictive performance.
  - R^2 is useful for comparing models, but it doesn't guarantee a good fit—always combine it with residual analysis.

OBJECTIVE AND CONSTRAINT FUNCTION FOR N-TORC

$$\text{Minimize:} \quad \sum_{i \in \text{layers}} \left(\widehat{\text{LUTS}}_i + \widehat{\text{FF}_i} + \widehat{\text{BRAM}}_i + \widehat{\text{DSP}_i} \right)$$

$$\text{Subject to:} \quad \sum_{i \in \text{layers}} \widehat{\text{latency}}_i \leq 50000$$

TRAINING AND DEPLOYMENT RESULTS FOR PARETO OPTIMAL NETWORKS

- Network sizes range from 12K total multiplies to 75K which is tiny compared to potential range

- This indicates large potential for improving accuracy in this range but no benefit beyond a 75K model.

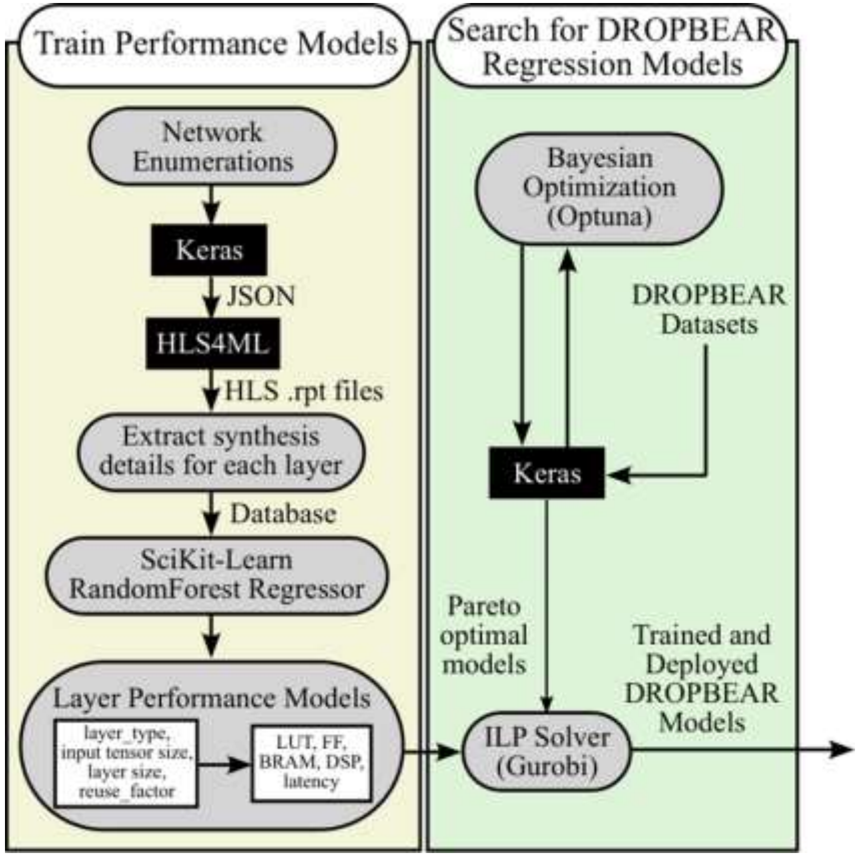
Accuracy (RMS error)	Workload (Multiplies)	# LUTS	# DSPs	Latency (µs)	Optimized RF for Each Layer
0.169	11.9K	18999	10	168.83	48, 768, 384, 768, 384, 64
0.1433	12.2K	24808	17	169.14	48, 384, 384, 384, 768, 64, 16, 16, 16, 4
0.1339	12.3K	24807	17	169.14	48, 768, 768, 384, 768, 64, 25, 25, 25, 5
0.119	12.6K	24807	17	169.14	48, 384, 768, 384, 768, 512, 32, 32, 32, 4
0.1161	13.7K	26375	16	171.82	48, 768, 768, 768, 768, 384, 162, 162, 18
0.1134	15.7K	26375	16	171.82	48, 768, 768, 768, 768, 384, 162, 162, 18
0.1095	16.8K	27125	14	171.82	60, 600, 1200, 300, 1200, 1360, 289, 289, 17
0.1065	21.7K	63052	40	193.92	78, 2028, 1014, 2028, 2028, 1768, 289, 289, 17
0.1029	25.0K	63052	40	193.92	90, 2700, 2700, 2700, 2700, 2040, 289, 289, 17
0.0982	25.6K	30836	24	170.59	24, 192, 384, 768, 384, 1824, 1444, 38
0.0958	33.0K	44702	30	176.81	24, 192, 384, 384, 768, 4512, 2209, 2209, 2209, 2209, 47
0.0939	34.4K	63052	40	194.94	123, 5043, 5043, 5043, 5043, 3116, 361, 361, 19
0.0851	36.6K	80227	58	174.88	24, 192, 768, 768, 384, 5600, 2500, 2500, 2500, 50
0.0828	41.4K	91708	66	176.96	24, 192, 768, 768, 768, 336, 2916, 2916, 2916, 2916, 54
0.0813	70.5K	91702	66	176.96	24, 192, 768, 768, 768, 13200, 5625, 5625, 5625, 5625, 75
0.0792	74.9K	94960	78	193.26	24, 192, 192, 192, 768, 14592, 5776, 5776, 5776, 5776, 76

N-TORC COMPARISON WITH DIFFERENT DSE

Network	Trials	Stochastic Search				Simulated Annealing (SA)				N-TORC			
		# LUTs	# DSP	Latency (µs)	Search Time (s)	# LUTs	# DSP	Latency (µs)	Search Time (s)	# LUTs	# DSP	Latency (µs)	Search Time (s)
Model 1	1K	137034	209	124	5	120481	159	111	4	94960	78	193	5
	10K	106522	134	189	47	104306	101	162	38				
1.3e11 RF	100K	100054	107	140	413	98289	101	156	382				
permuations	1M	95537	79	192	4573	93046	136	193	3995				
Model 2	1K	445328	746	190	6	434219	720	162	6	374009	459	199	6
	10K	415243	646	198	53	398131	576	196	56				
3.4e11 RF	100K	391543	508	191	565	396019	514	187	567				
permuations	1M	383849	474	190	5406	376416	466	196	4694				

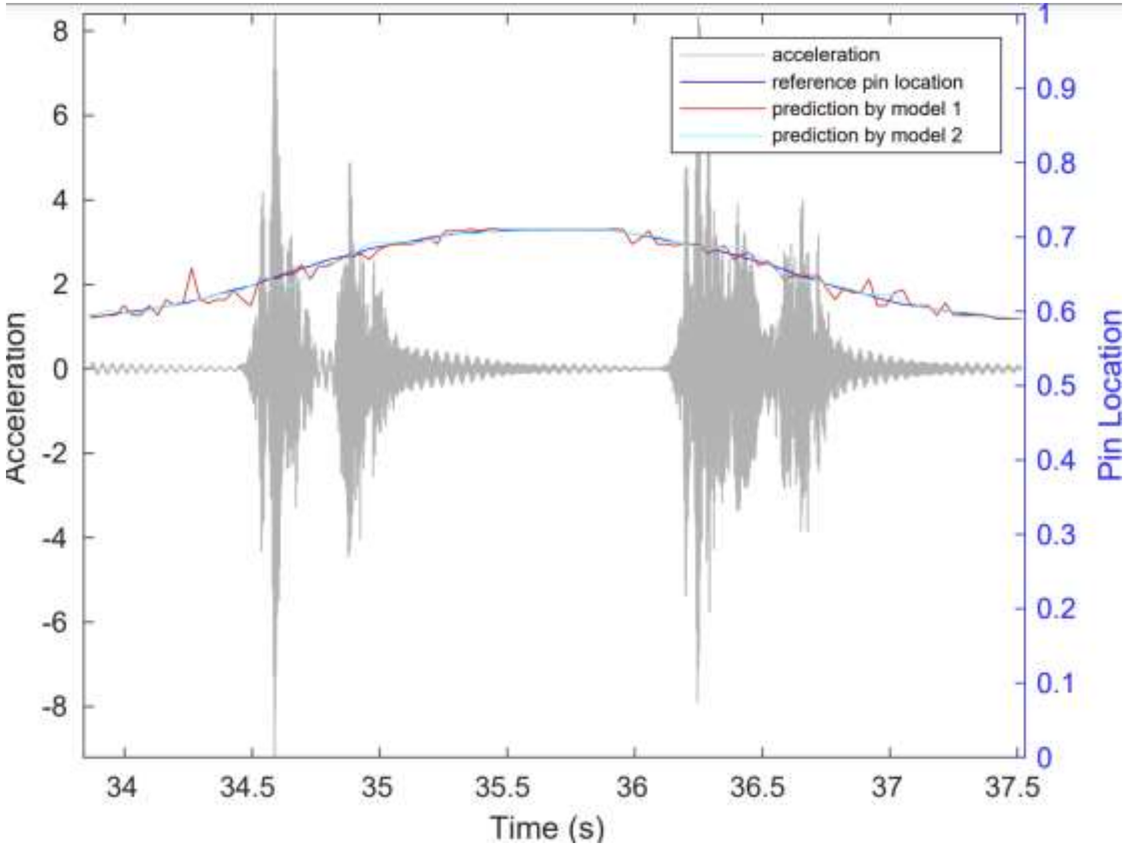
TOOL FLOW OVERVIEW

- First Stage : Bayesian Optimiztion "Optuna"
- Second Stage: Integer Linear Program Solver "Gurobi".



Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA

CONCLUSION AND FUTURE WORK



- It supports different data precision on each layer .
- Currently we support 16 bits on each layer
- We are working on incorporating varying precisions into our performance and cost models.

THANKS!
Q&A

Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA