

Muhammad Ali Farooq<sup>1</sup>, Abid Rafique<sup>2</sup>, Suhaib A. Fahmy<sup>3</sup>, Aman Arora<sup>1</sup>

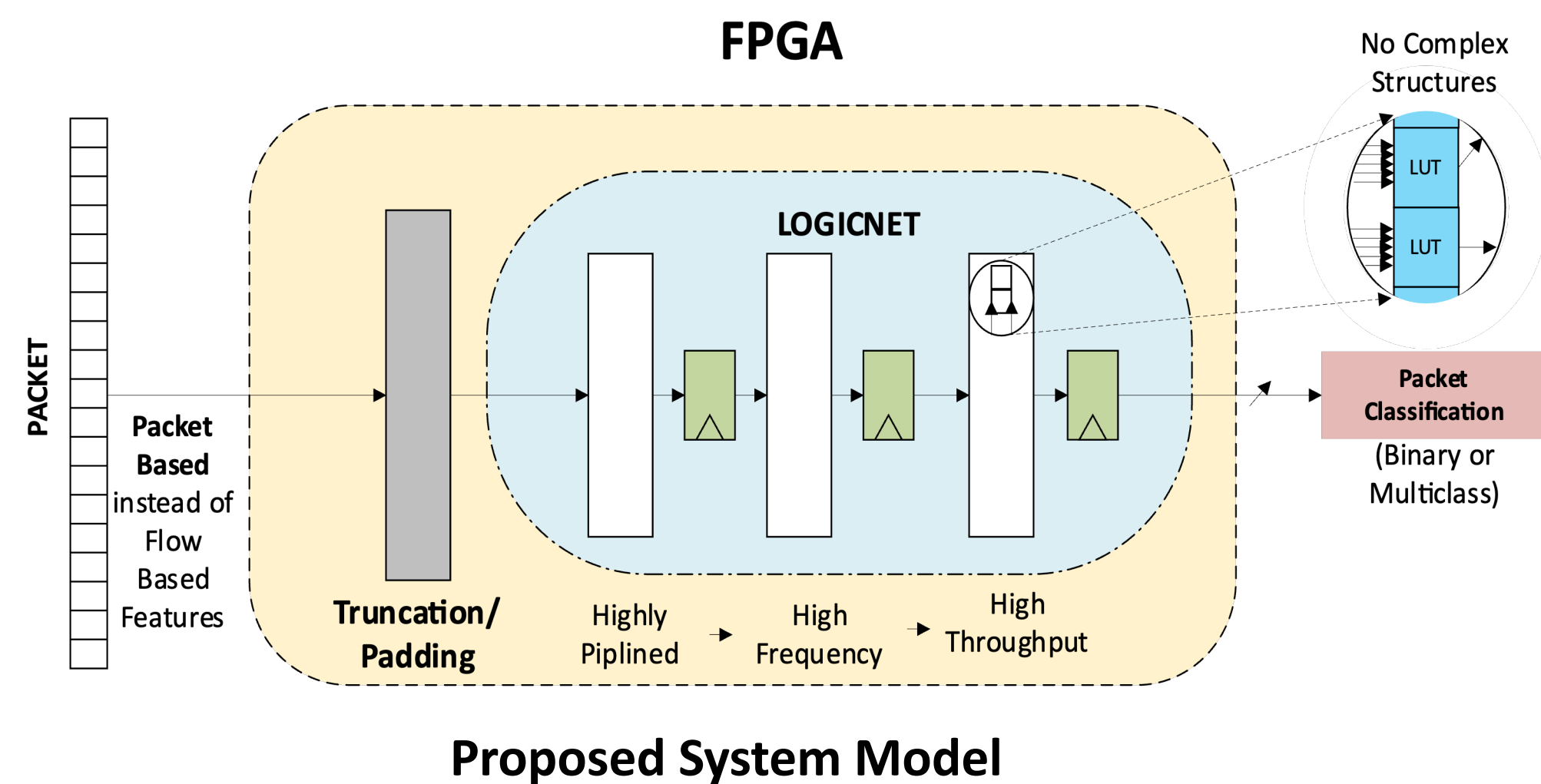
<sup>1</sup>Arizona State University, <sup>2</sup>National University of Sciences & Technology, <sup>3</sup>King Abdullah University of Science and Technology

## Introduction

- Network Intrusion Detection Systems (NIDS) are critical for monitoring and securing modern networks.
- Traditional ML-based NIDS require pre-processed flow features, which introduce computational overhead and latency.
- FPGAs offer high-throughput and low-latency computation, making them suitable for real-time NIDS deployment.
- Challenges of FPGA-based NIDS:
  - Reliance on complex feature extraction reduces real-time feasibility.
  - Implementing large ML models on resource-constrained FPGAs is difficult.

## Proposed Solution

- We propose a novel FPGA-accelerated NIDS that directly classifies raw packets, eliminating feature extraction.
- Single-packet classification approach: Uses first 64 bytes of a packet to make real-time decisions.
- LUT-based neural network: Optimized for high-speed, low-power FPGA implementation.



## Methodology

- Edge-IIoT Dataset Used
- Dataset Preparation:

Truncating packets in the .pcap files to 64 bytes using Wireshark

Extracting byte-level information from the truncated packets and converting to plaintext

Cleaning the text files and converting the data from hexadecimal to binary

Padding packets smaller than 64 bytes for uniformity

Adding labels to indicate the traffic class

Removing duplicate packets to ensure data integrity

Randomly split the dataset into train, validation, and test sets

- Models developed for binary, 6-class and 15-class classification
- Neural Architecture Search performed to select optimal models
- LogicNets Framework used for model training and synthesis
- Synthesis performed for Virtex Ultrascale+

## Architecture Search

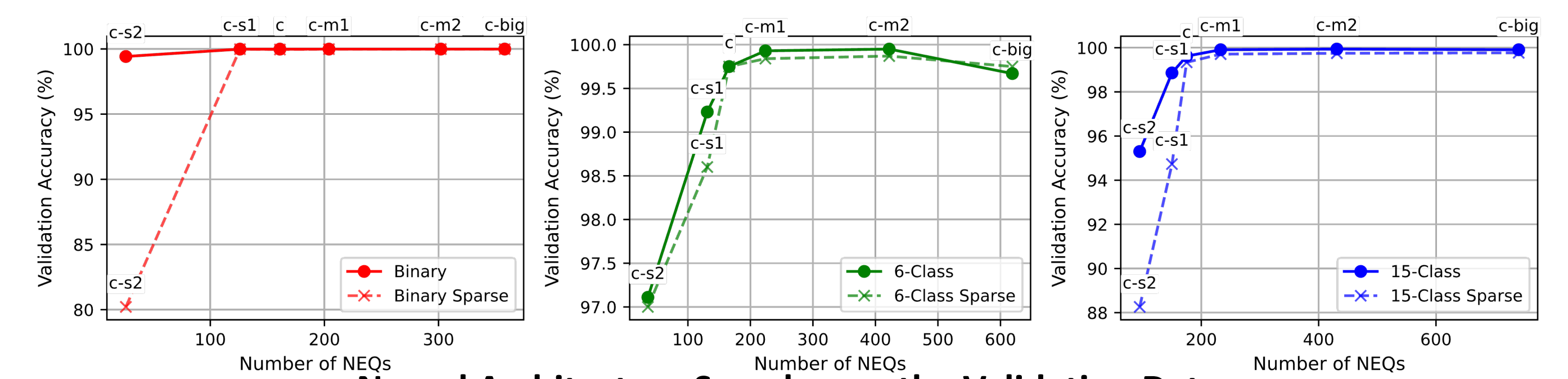
Model	Binary	6-Class	15-Class
c-s2	25,1	30,6	80,15
c-s1	100,25,1	100,25,6	100,25,15
c	128,32,1	128,32,6	128,32,15
c-m1	128,60,15,1	128,60,30,6	128,60,30,15
c-m2	200,86,15,1	300,86,30,6	300,86,30,15
c-big	256,86,15,1	512,86,15,6	593,100,33,15

**Model Definitions:** Each value represents the number of Neuron Equivalents (NEQs) in a single layer

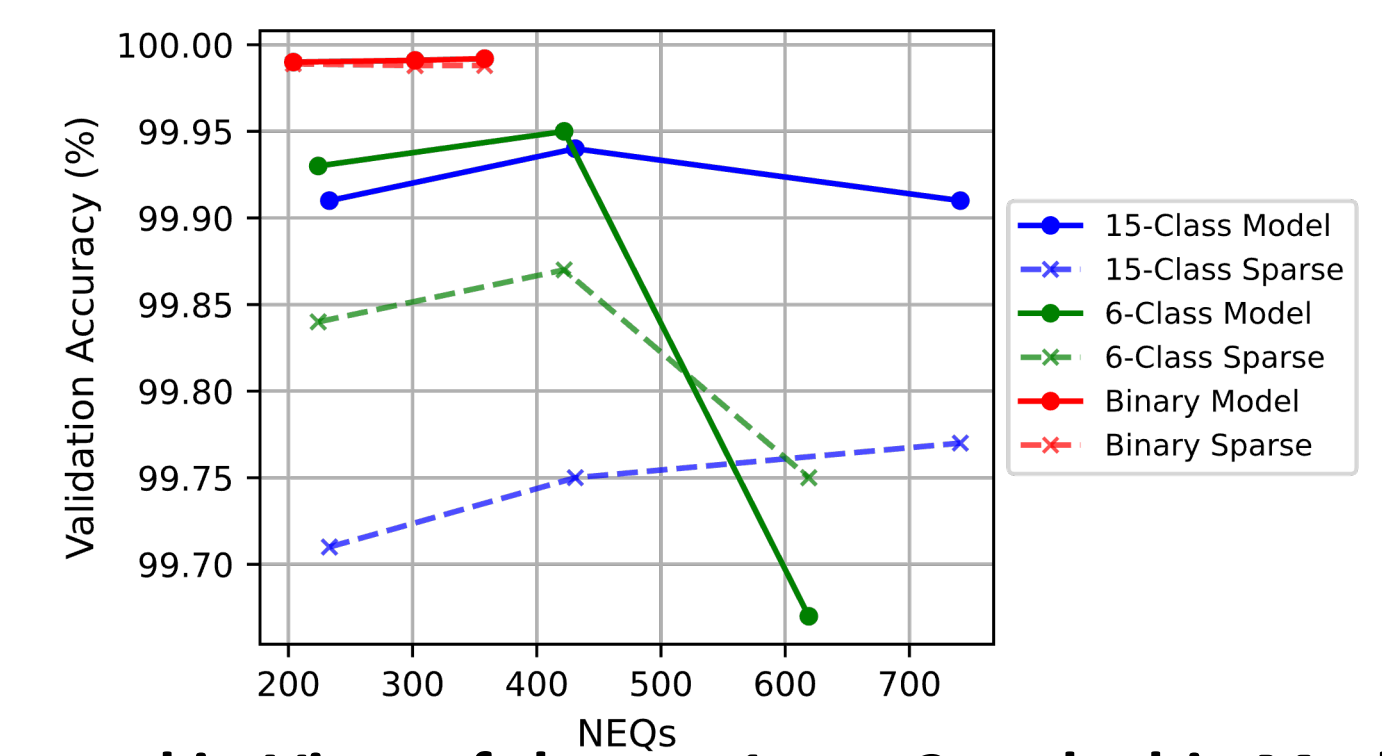
	$\beta_i$	$\gamma_i$	$\beta$	$\gamma$	$\beta_o$	$\gamma_o$
Regular	1	6	2	6	2	7
Sparse	1	4	2	4	2	7

**Sparsity Definitions:** The two sparsity settings explored.  $\beta$  refers to the bitwidth of each input to an NEQ, and  $\gamma$  refers to the number of inputs to an NEQ. Columns marked with  $i$  refer to the input layer and  $o$  refer to the output layer

- We consider a regular (non-sparse) and sparse model for each model architecture.
- Sparse models differ from their non-sparse counterparts in the  $\gamma$  value, but not in the  $\beta$  value.
- Increased sparsity means fewer connections between NEQs and allows the creation of resource-efficient implementations
- For 6-class and 15-class models, accuracy even declines after a certain size, making further increase in model size unnecessary.
- For binary models, the accuracy gain beyond a certain point is minimal, making further increases in model size inefficient.



Neural Architecture Search over the Validation Data



Zoomed in View of the c-m1, c-m2 and c-big Models

## Results

### Machine Learning Performance:

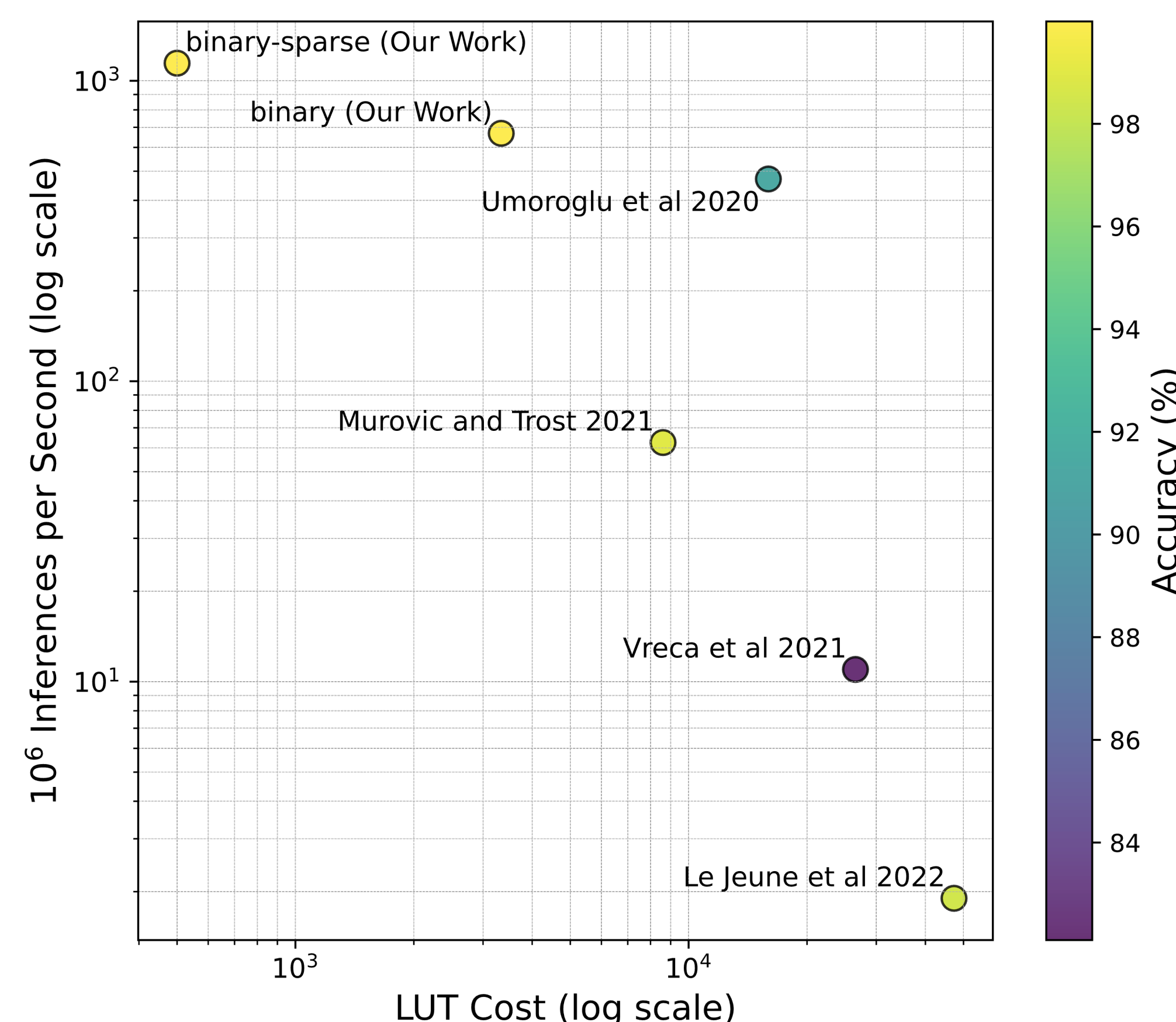
- Binary classification accuracy: **99.99%** (Sparse: **99.99%**)
- 6-class accuracy: **99.95%** (Sparse: **99.87%**)
- 15-class accuracy: **99.95%** (Sparse: **99.75%**)

### FPGA Implementation Results:

- All models use **<10,000 LUTs**.
- Sparse models achieve **>1100 million inferences per second**.
- Low latency (**<7 ns**), ensuring real-time performance.

	ML Performance on Test Set			FPGA Implementation				
	Acc (%)	F-1 (%)	DS (%)	LUT	FF	Latency (ns)	Throughput	$f_{max}$ (MHz)
binary	99.994	99.994	99.994	3337	1496	5.984	668.45	668.45
binary-sparse	99.992	99.992	99.992	500	337	3.496	1144.16	1144.16
6-class	99.95	99.95	99.995	7033	2319	6.136	651.89	651.89
6-class-sparse	99.87	99.88	99.982	1652	736	3.44	1162.79	1162.79
15-class	99.95	99.95	99.989	9033	2455	5.863	681.66	681.66
15-class-sparse	99.76	99.79	99.964	2764	861	3.528	1133.79	1133.79

## Comparison with State-of-the-Art



## Discussion and Conclusion

- Achieves high accuracy (**>99.9%**) with minimal hardware costs.
- Sparse models provide massive improvements in efficiency, achieving **>1100 million inferences per second**
- Advantages of LUT-based NIDS:
  - Eliminates feature extraction overhead → Enables real-time inference.
  - Sparse architectures optimize FPGA resource usage → Suitable for resource-constrained environments.
  - Single-packet classification → Provides ultra-fast response time.
- Limitations:
  - Single-packet analysis lacks context → Cannot detect multi-step attacks
- Future Work:
  - Hybrid approaches needed. Need to combine single-packet classification with flow-based analysis.
  - Integrate rule-based anomaly detection for enhanced security.