

An Exploration of Machine Learning Cryptanalysis of a Quantum Random Number Generator

Abenezer Monjor
Division of Science and Mathematics
University of Minnesota,
Morris, Minnesota, USA



November 16, 2019



Machine Learning Cryptanalysis of a Quantum Random Number Generator

Nhan Duy Truong¹, *Student Member, IEEE*, Jing Yan Haw, Syed Muhamad Assad, Ping Koy Lam, and Omid Kavehei², *Senior Member, IEEE*

Abstract—Random number generators (RNGs) that are crucial for cryptographic applications have been the subject of adversarial attacks. These attacks exploit environmental information to predict generated random numbers that are supposed to be truly random and unpredictable. Though quantum random number generators (QRNGs) are based on the intrinsic indeterministic nature of quantum properties, the presence of classical noise in the measurement process compromises the integrity of a QRNG. In this paper, we develop a predictive machine

based on deterministic algorithms and can exhibit long-range correlation. For instance, weak cryptographic keys due to the poor source of randomness have been a known threat for years [2], [3]. In 2012, the biggest scan of Transport Layer Security (TLS) and Secure Shell (SSH) at the time unveiled surprisingly widespread vulnerable keys [2]. In fact, Heninger *et al.* [2] managed to acquire private keys for 0.5%

Outline

1. Background
2. Paper Details
3. Result
4. Conclusion

1. Background

- Randomness
- Probability Distribution
- Entropy
- Random Numbers Generators
- Correlation and Autocorrelation
- Neural Network and Deep Learning

1. Background: Randomness

Lack of predictability in a sequence of events

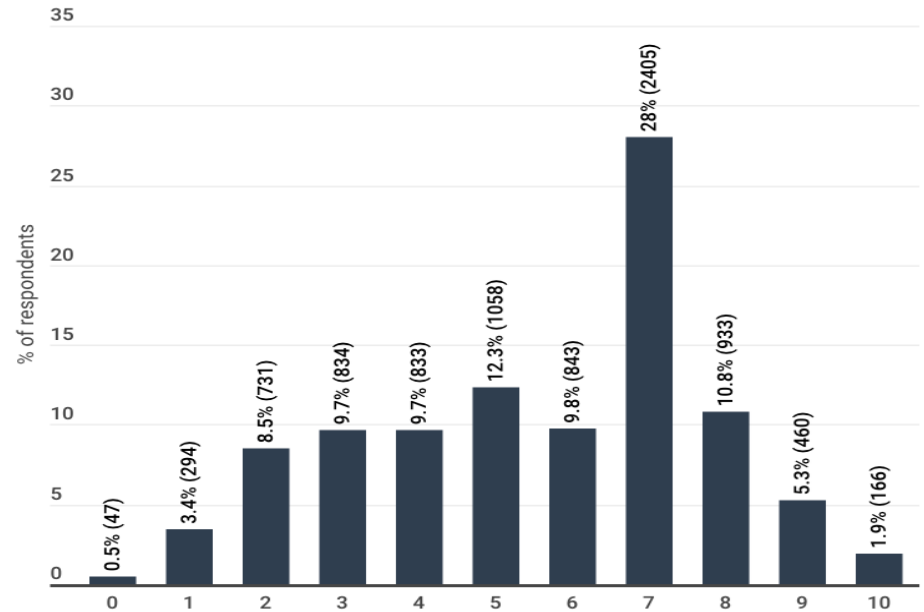
1. Background: Randomness

Pick a random
number
between 1 and
10?

1. Background: Randomness

Pick a random number between 1 and 10?

Pick a random number from 1-10
($n=8604$, $mean=5.687$, $median=6$)



The probability distribution of random numbers between 1 and 10.

1. Background: Probability Distribution

- List of all possible outcome of random variable with probability value
- Can be expressed in the form of table, a graph

1. Background: Entropy

- Measure of the disorder

$$S = - \sum_i P_i \log P_i$$

- Maximum when all the outcome are equally likely
- Always greater than or equals to 0

1. Background: Randomness

Classical:

- Deterministic
- Based on lack of knowledge
- Not-truly random
Eg. Coin flipping

Non-classical:

- Non-deterministic
- Based on non-deterministic physical process
- Truly random
Eg. Radioactive materials decays

1. Background: Random Numbers Generators

- Generate a sequence of random numbers
- Used in a wide array of application :
 - Gaming
 - Simulation
 - Cryptography

1. Background: Random Numbers Generators

Pseudo-Random Number Generator

- Deterministic
- Uses software algorithm
- Determined by initial input - seed
- Same seed = same sequence

$$X_{n+1} = (aX_n + c) \bmod m$$

where X is the sequence of pseudo-random values

m , $0 < m$ - modulus

a , $0 < a < m$ - multiplier

c , $0 \leq c < m$ - increment

x_0 , $0 \leq x_0 < m$ - the seed or start value

Eg. Linear Congruential Generator(LCG)

1. Background: Random Numbers Generators

Hardware Random Number Generator

- Non-deterministic
- Generates sequences using physical process
- Same input = Different output
- No discernable pattern
- Best suited for encryption key generation

1. Background: Correlation & Autocorrelation

Correlation:

- Measures linear relationship between variables

Autocorrelation:

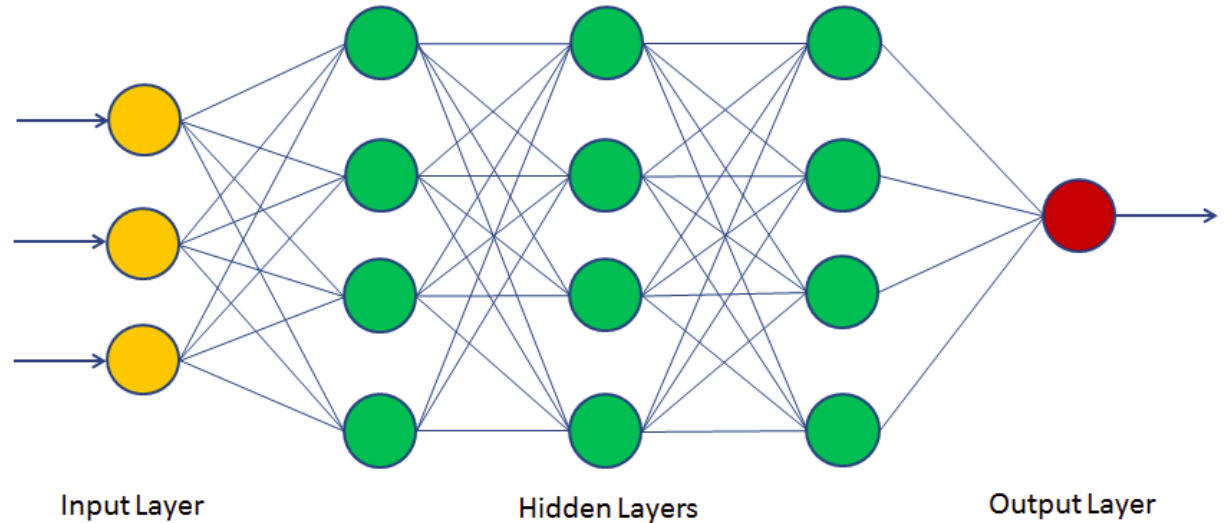
- Lagged correlation
- Compares changing variable to itself at different time points

1. Background: Neural Network and Deep Learning

- Inspired by structure of our brain
- Designed to recognize patterns in data
- Can guess next generated random numbers using previous values

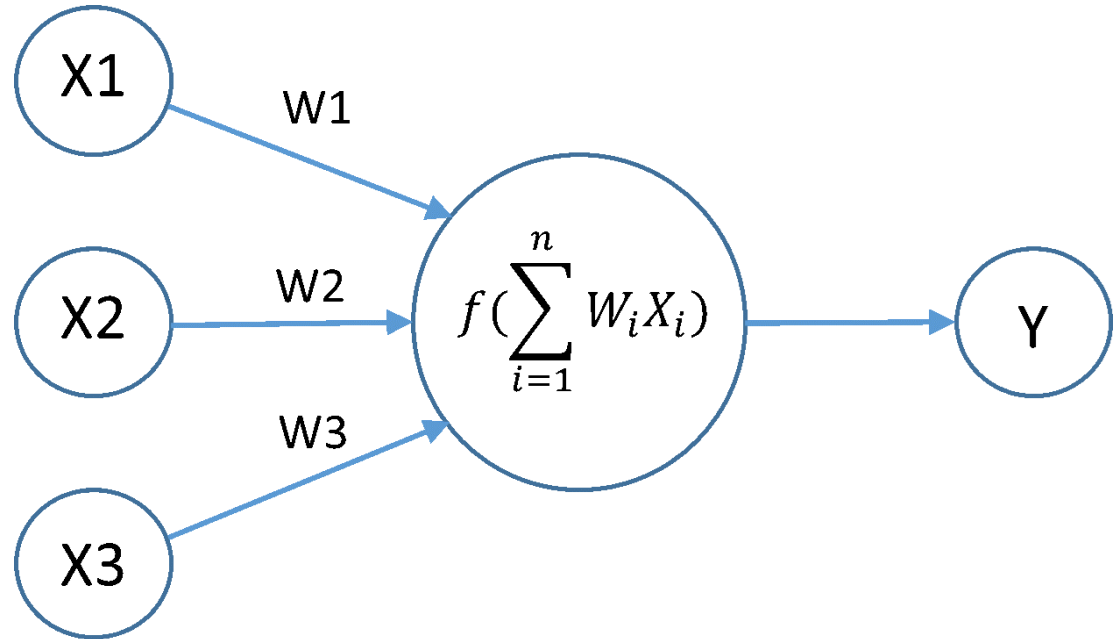
1. Background: Neural Network and Deep Learning

- Input Layer
- Hidden Layers
- Output Layer





1. Background: Neural Network and Deep Learning

- Inputs
- Weights
- Activation Function
- Output



1. Background: Neural Network and Deep Learning

- Predication: 

```
graph LR; Input[Input] --> NN[NN]; NN --> Predication[Predication]
```
- Training 

```
graph LR; Weight[Weight] --> NN[NN]; NN --> Cost[Cost]
```
- Cost compare prediction to target
- In paper called **label**

1. Background: Training

- Improving predictions
- Adjusting weights
- By comparing prediction to label
- Usually data set is split into two: for training and testing purposes

1. Background: Training

- Training set run through the network
- Updates weights
- Repeat until predicted outputs close to labels

1. Background: Testing

Used to evaluate the network

1. Background: Overfitting

- Network learns the training data too well
- Network performs well on the training set but performs poorly on testing set

1. Background: One Hot Encoder

- Representation of categorical variable good for ML
- 0 indicates non-existent
1 indicates existent

color	color_red	color_blue	color_green
red	1	0	0
green	0	0	1
blue	0	1	0
red	1	0	0

1. Background: Recurrent Convolutional Neural Network (RCNN)

- Combination of convolutional neural network and recurrent neural network

1. Background: Convolutional Neural Network (CNN)

- Well known for its use in analyzing visual imagery
- Inspired by the visual cortex
- Kernels detect patterns



1. Background: Convolutional Neural Network (CNN)

Kernel:

- Small matrices of weights
- Slides over the input data

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

1. Background: Recurrent Neural Network (RNN)

- Used to find potential patterns of a long sequences of data
- Generates outputs using current input and the previous computation

1. Background: Long Short Term Memory (LSTM)

- One of RNN's architecture
- Capable of learning long-term dependencies
- Effective addressing sequence to sequence problem

1. Background: Recurrent Convolutional Neural Network (RCNN)

- A combination of RNN and CNN
- Implemented by feeding features extracted by CNN into RNN

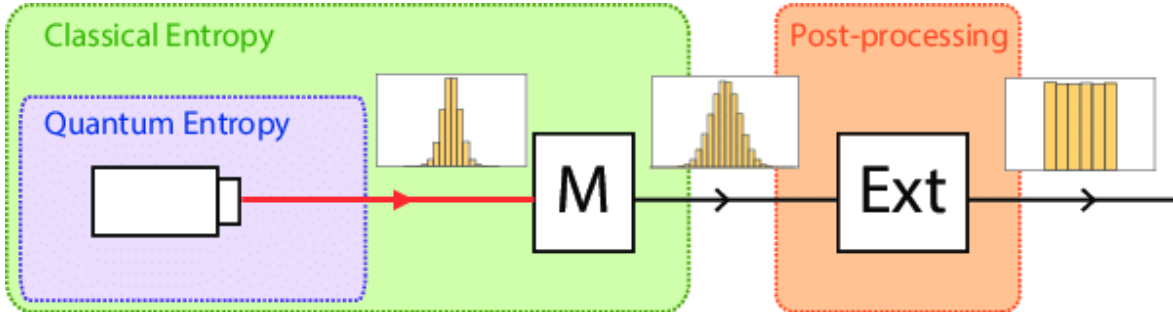
Outline

1. Background
2. Paper Details
3. Result
4. Conclusion

2. Paper Details

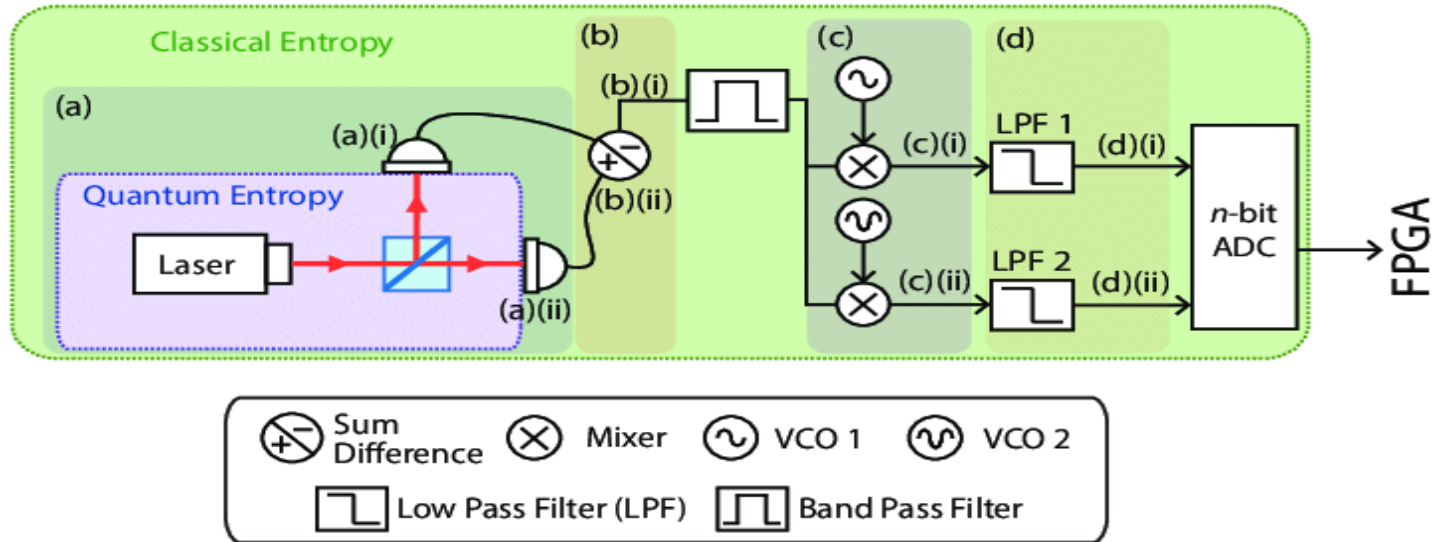
- Investigate how much classical entropy affects randomness of QRNG
 - First scenario: classical noise
 - Second scenario: quantum and classical noise
- Apply ML based predictive analysis on different stages of the QRNG

QRNG Setup



- Two segments:
 - Entropy source
 - Post-processing procedures
- Entropy source = Produce raw randomness
- Post-processing block = Extract quantum randomness

QRNG Setup

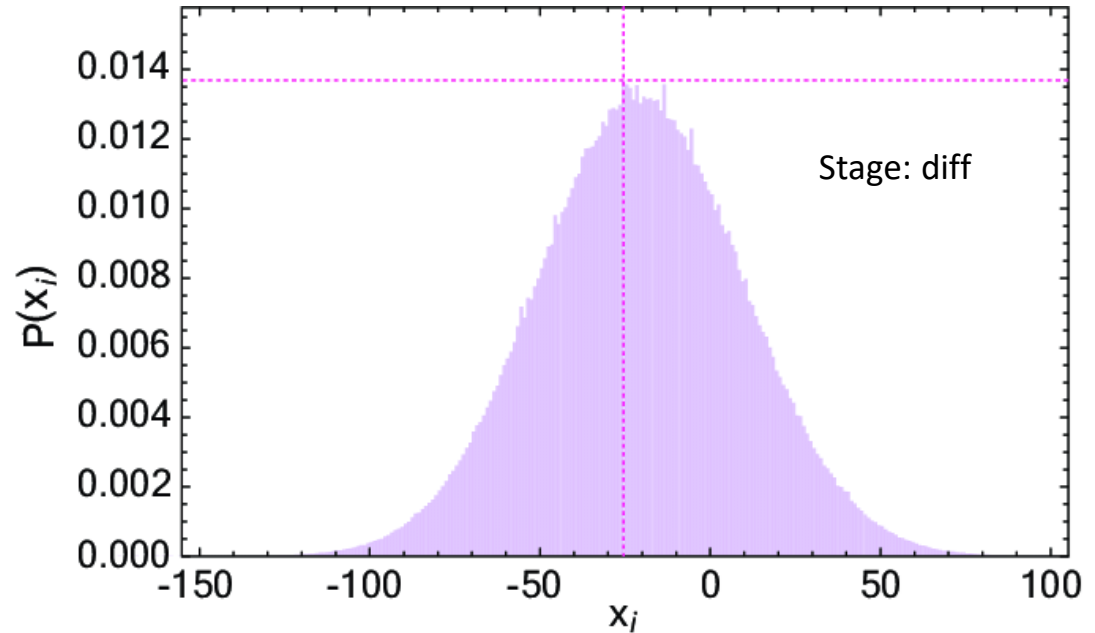


Data acquisition stages in the entropy blocks of the QRNG. Stage (a): (i) detector 1 and (ii) detector 2; Stage (b): (i) difference and (ii) sum of the photocurrents; Stage (c): difference of the photocurrents demodulated at (i) 1.375 GHz (ii) 1.625 GHz; Stage (d): Low pass filtering of the signals from (c).

2. Paper Details

- By using ML they are finding potential patterns generated numbers at different stages of the QRNG.
- Probability of successful output prediction = P_{ml}
- Guessing probability = P_g

2. Paper Details



- Best strategy to guess the value of -26 , giving a success rate of 1.37%

2. Paper Details: Data set

- 10 million 16-bit integer(turned into 13 bit integers)
- First 5 million data used as training set
- Remaining 5 million data is divided into 5 test-sets each of which contains 1 million data point

2. Paper Details: Data Preparation

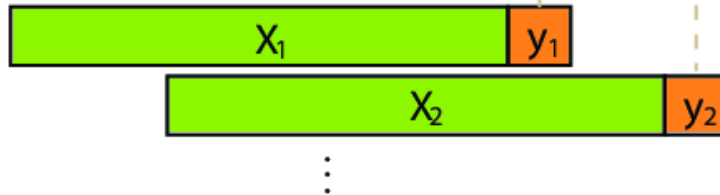
- By using the previously generated numbers predict the next number at each stages of the QRNG.
- 100 adjacent numbers are considered as input and the next number is considered as the label.

2. Paper Details: Data Preparation

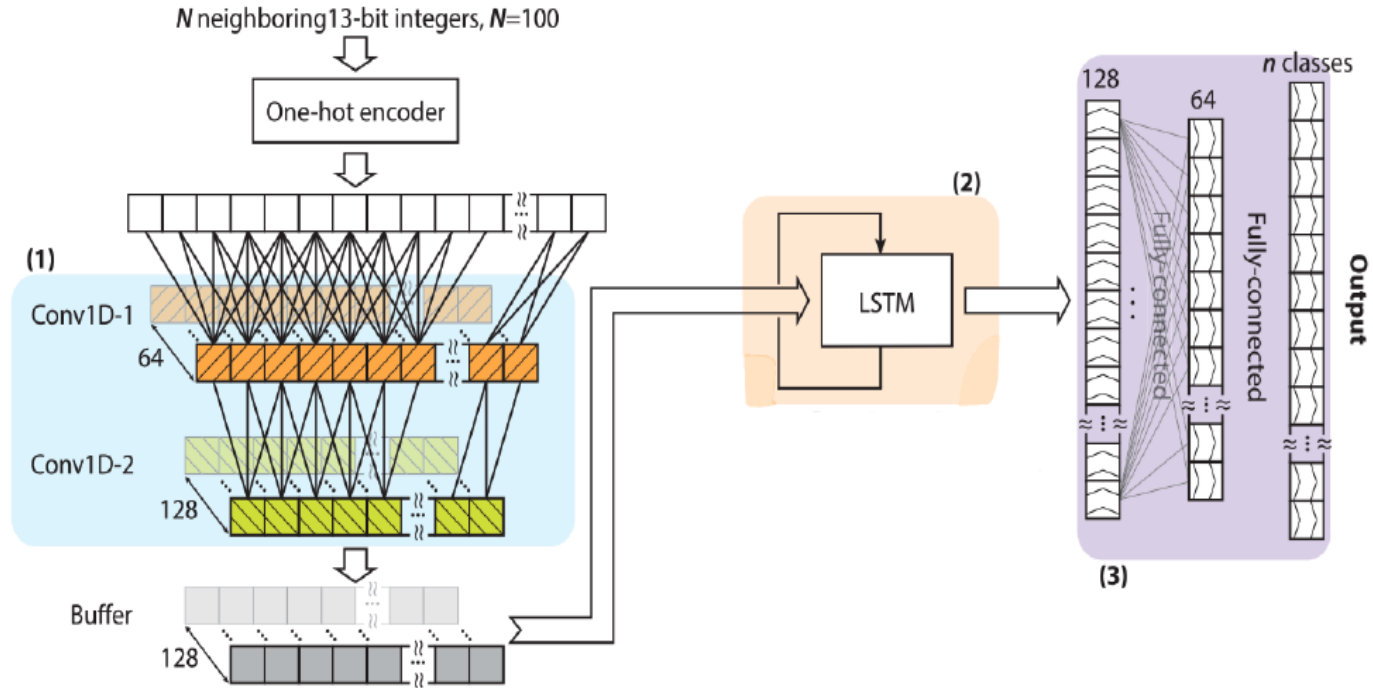
- S is used to control the overlap between samples.
- $N = 100$ and $S = 3$



$a_1, a_2, a_3, a_4, a_5, a_6, \dots, a_N, a_{N+1}, \dots, a_{N+S+1}, \dots$



2. Paper Details



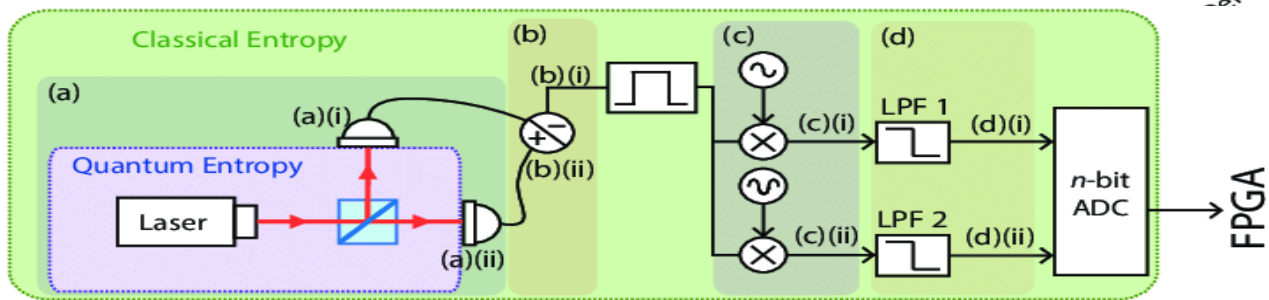
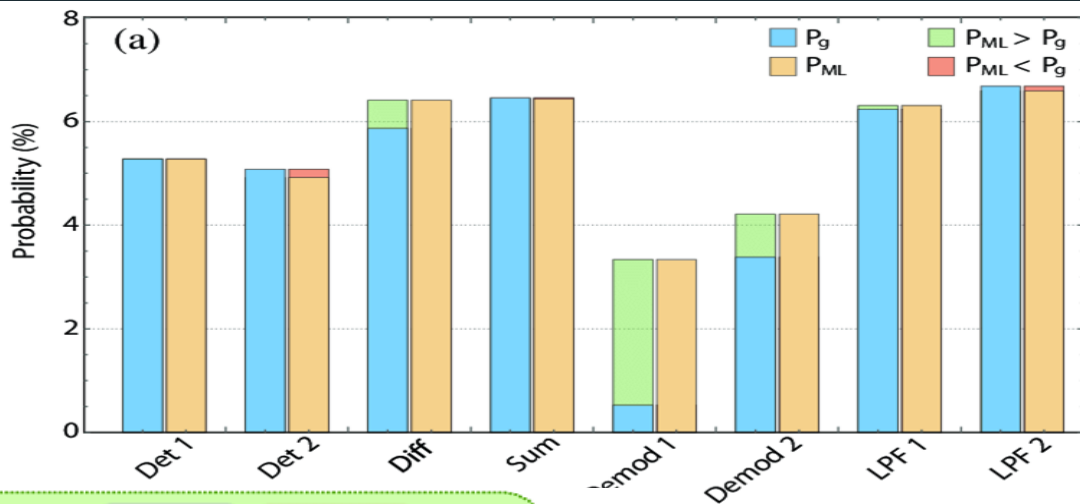
Outline

1. Background
2. Paper Details
- 3. Result**
4. Conclusion

3. Results

Classical entropy:

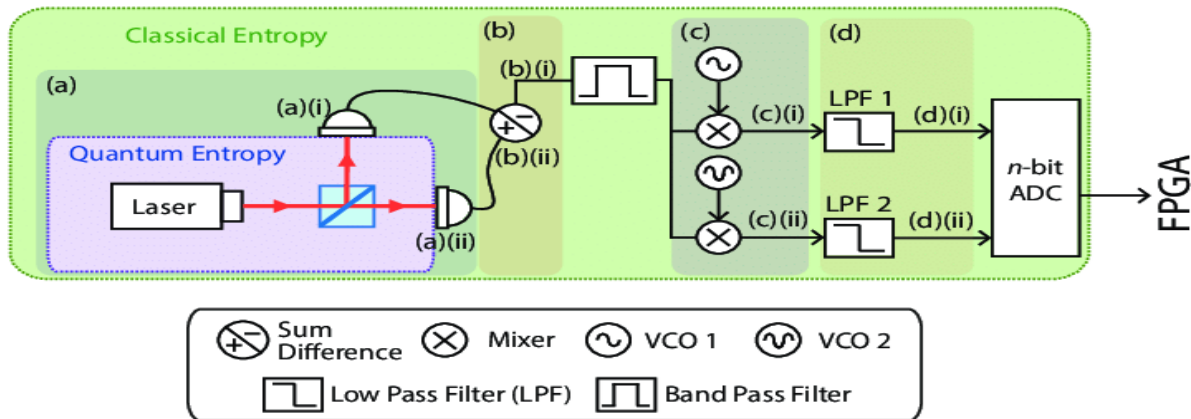
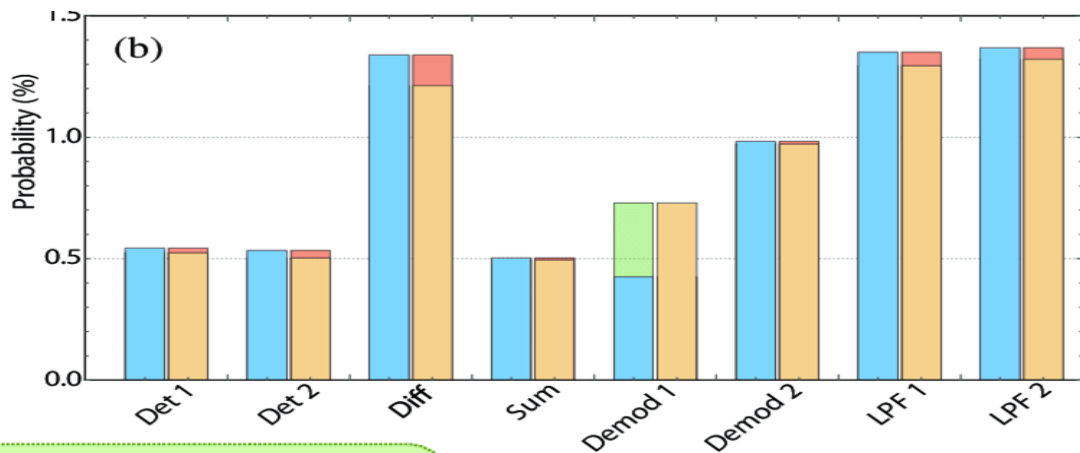
- RCNN model shows $P_{ml} > P_g$ in 4 out of 8 stages.



3. Results

Classical and Quantum Entropy:

- $P_{ml} > P_g$ in $1/8$ stages.
- Considerable decrease



3. Results

- Probability Distribution

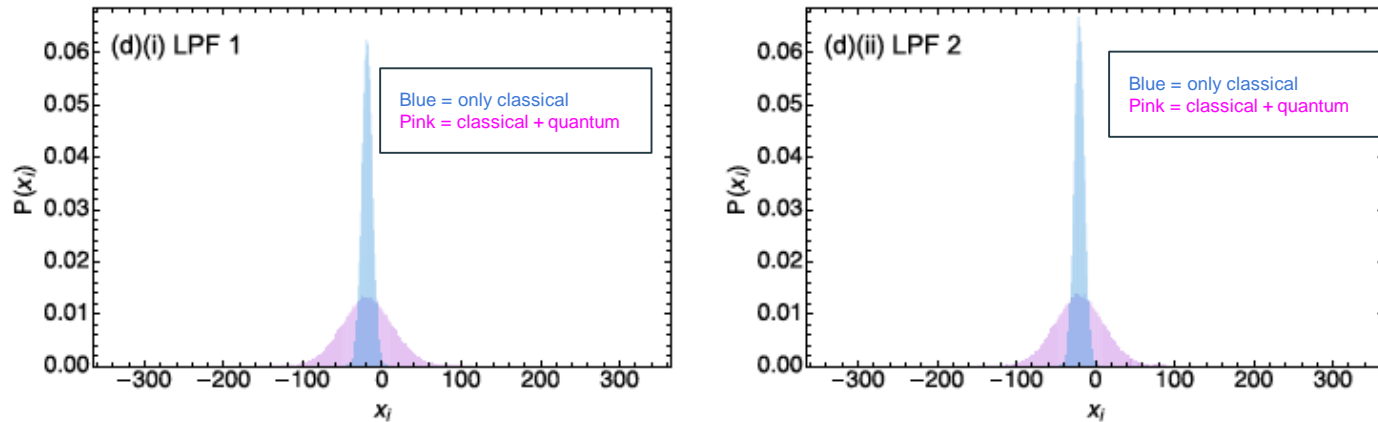
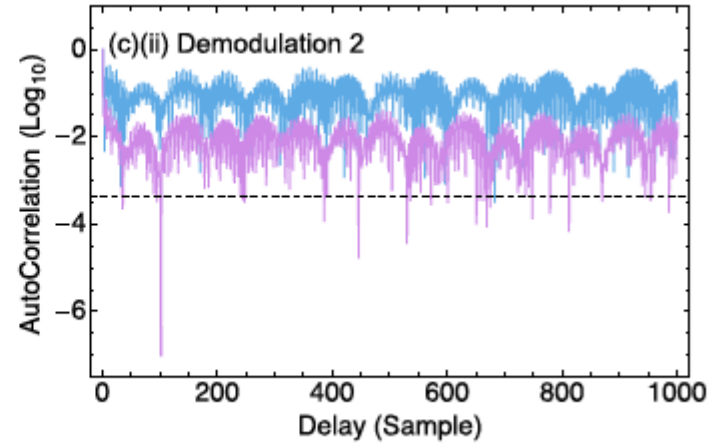
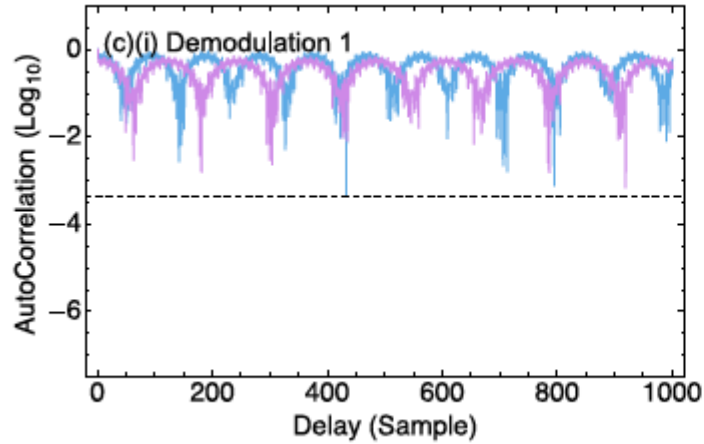


Fig. 8. Distribution of datasets in scenario 1 (blue), where only electrical noise is present and scenario 2 (pink), where both the electrical and quantum noise are present. $SD_{M(E)}$ is the standard deviation of the measured (electronic) signal.

3. Results

- Autocorrelation



3. Results

- Autocorrelation

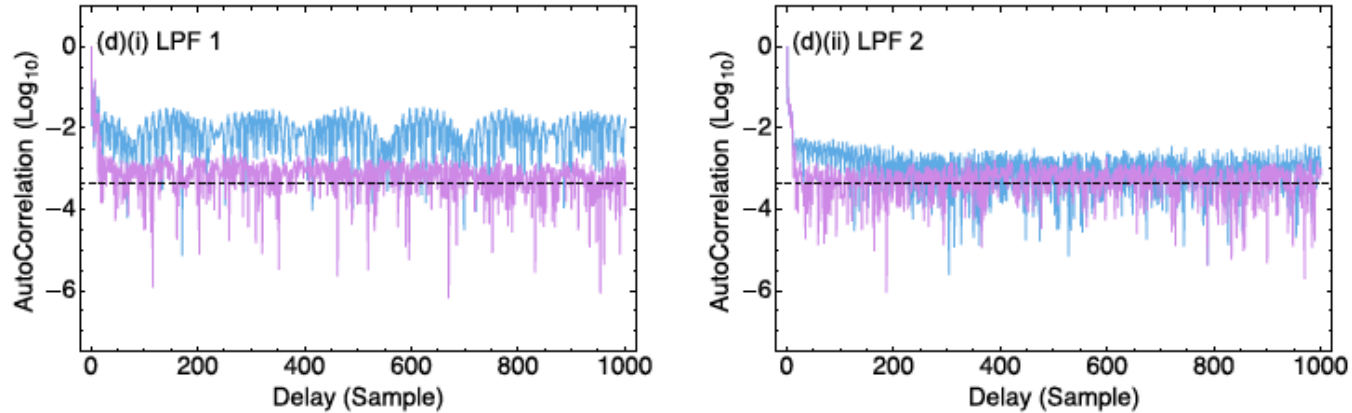
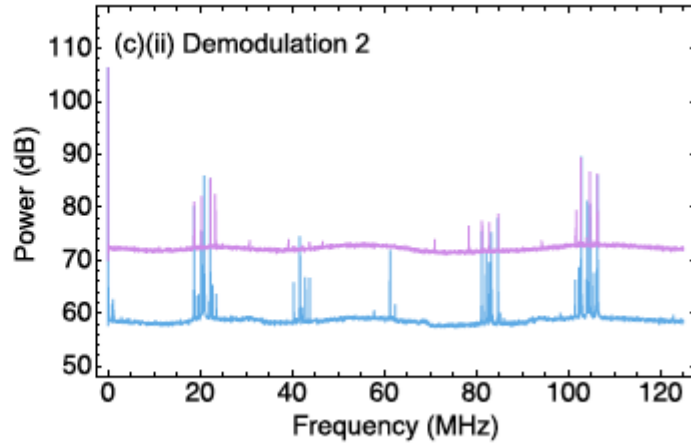
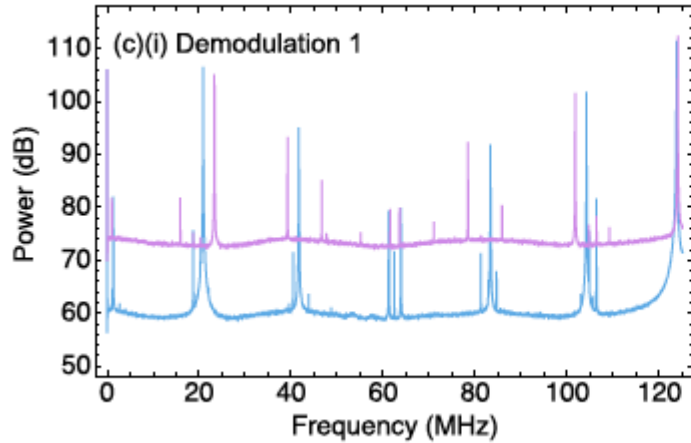


Fig. 9. Autocorrelation of datasets in scenario 1 (blue), where only electrical noise is present and scenario 2 (pink), where both electrical and quantum noise are present. Dashed lines show the theoretical standard deviation of truly random 5 million samples.

3. Results

- Power Spectral Density (PSD)



Outline

1. Background
2. Paper Details
3. Result
4. Conclusion

4. Conclusions

- Huge decrease in ML accuracy when both classical and quantum noise
- Did not explore how knowledge from one stage gives knowledge about another stage

Acknowledgment

Thanks for your time and attention!

Thank you to my advisor Peter Dolan for guidance and feedback.

References

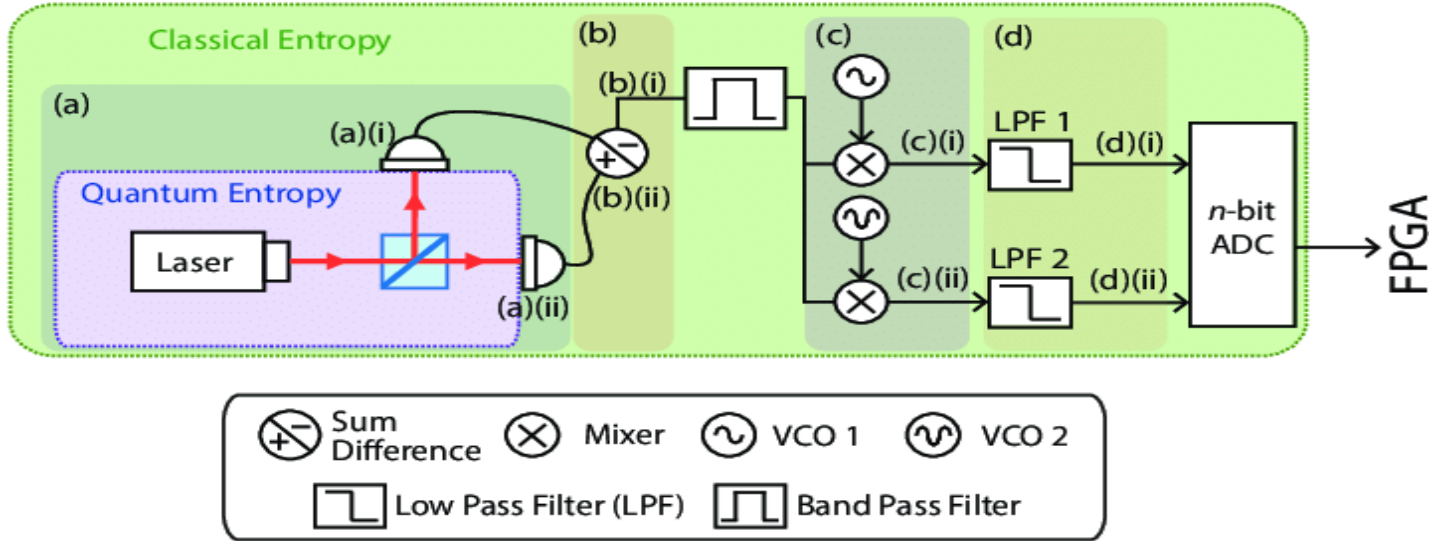
- [1] M. Herrero-Collantes and J. C. Garcia-Escartin. Quantum random number generators. *Reviews of Modern Physics*, 89(1), Feb 2017.
- [2] M. Stipcevic. Quantum random number generators and their applications in cryptography. *Advanced Photon Counting Techniques VI*, May 2012.
- [3] N. D. Truong, J. Y. Haw, S. M. Assad, P. K. Lam, and O. Kavehei. Machine learning cryptanalysis of a quantum random number generator. *IEEE Transactions on Information Forensics and Security*, 14(2):403–414, Feb 2019.

Image References

- https://twitter.com/bill_gross/status/1086619234426413056
- “Deep Neural Network's Precision for Image Recognition, Float or Double?” Stack Overflow stackoverflow.com/questions/40537503/deep-neural-networks-precision-for-image-recognitionfloat-or-double.”<https://stackoverflow.com/questions/40537503/deep-neural-networks-precision-for-image-recognitionfloat-or-double>
- “Identifying Road Signs Using a Convolutional Neural Network”<https://medium.com/@zsyed350/identifying-road-signs-using-a-convolutional-neural-network-1c9ec9b654c9>
- “Neural Network Models in R”
<https://www.datacamp.com/community/tutorials/neural-network-models-r>
- “Entropy(Information theory)”[https://en.wikipedia.org/wiki/Entropy_\(information_theory\)](https://en.wikipedia.org/wiki/Entropy_(information_theory))

Questions?

3. Results



Data acquisition stages in the entropy blocks of the QRNG. Stage (a): (i) detector 1 and (ii) detector 2; Stage (b): (i) difference and (ii) sum of the photocurrents; Stage (c): difference of the photocurrents demodulated at (i) 1.375 GHz (ii) 1.625 GHz; Stage (d): Low pass filtering of the signals from (c).

3. Results

- Probability Distribution

