Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Neural Network Enhanced Blockchain Mining: A Modular-Exponentiation Proof-of-Work Framework with Learned Nonce Prediction

Fiza Shaikh

Department of Computer Science, Atlantic Technological University, Ireland

This work was supported in part by the Department of Computer Science, Atlantic Technological University, Ireland.

ABSTRACT This paper presents Modular-Exponentiation Proof-of-Work (MEPoW), a formally defined consensus puzzle in which a valid block requires a nonce n such that SHA-256(n || header) satisfies a leading-zero target and the block-derived tuple (a, b, m, T) satisfies a^b mod m = T (mod m). Because (a, b, m, T) are deterministically derived from the block header, the modular residue condition is learnable. A dense neural network trained on (a, b, m) predicts a proximity nonce, replacing blind iteration with a targeted residual search. We formally prove the derivation bijection, quantify prediction-to-nonce proximity, and show that model leakage does not weaken MEPoW security beyond the inherent speedup bound. Experimental evaluation on 600 blocks across difficulty levels 3-8 shows mining-time reductions from 8.4x to 15.0x over single-threaded CPU brute-force, with energy per block of 11.4 J vs. 446.7 J (CPU baseline) at difficulty 6. With GPU inference the neural miner achieves 0.77 J/block -- 580x below CPU baseline and 17x below GPU brute-force. Mean prediction relative error is 2.61%, and a hybrid verification layer maintains a 100% valid-block rate across 600 test blocks. Ablation studies and a structured five-vector security analysis are included.

INDEX TERMS : *Blockchain, energy efficiency, machine learning, MEPoW, mining optimisation, modular exponentiation, neural networks, proof-of-work.*

I. INTRODUCTION

A. Motivation and Scope

Proof-of-work (PoW) blockchain consensus [1] derives its security from computational hardness: finding a nonce that satisfies a hash target requires O(16^d) SHA-256 evaluations for d leading zero nibbles. SHA-256's resistance to pre-image attacks also makes it opaque to learned optimisation -- there is no exploitable algebraic structure for a neural predictor.

This motivates the central design choice: rather than claiming a neural network can predict SHA-256 outputs, we define MEPoW, whose validity condition has two independent components: (i) a standard leading-zero SHA-256 hash condition, and (ii) a modular residue condition a^b mod m = T (mod m) with parameters (a, b, m, T) deterministically derived from the block header. The residue condition is algebraically structured and therefore learnable.

MEPoW is intentionally a research puzzle rather than a production replacement for Bitcoin's PoW. Its value is as an internally consistent, formally analysable framework for studying the interaction between learned prediction and PoW efficiency.

B. Contributions

This paper makes the following contributions:

1) Formal MEPoW specification with proof that (a, b, m, T) are uniformly distributed under the random oracle model (Proposition 1).
2) Closed-form speedup bound S = 1/(2*e) proving savings increase monotonically with difficulty d (Proposition 2).
3) Neural architecture achieving mean relative prediction error of 2.61% on held-out test cases.
4) GPU baseline comparison: 0.77 J/block with GPU inference -- 580x below CPU baseline and 17x below GPU brute-force.
5) Five-vector security analysis proving model leakage does not weaken hash-based security.
6) Ablation study quantifying the independent contribution of each architectural component.

C. Paper Organisation

Section II reviews related work. Section III formalises MEPoW. Section IV describes the neural architecture. Section V covers implementation. Section VI presents results. Section VII provides the security analysis. Section VIII discusses limitations. Section IX concludes.

II. RELATED WORK

A. Proof-of-Work Variants

Nakamoto's SHA-256^2 PoW [1] underpins Bitcoin. Memory-hard variants include Scrypt [18] and Equihash [19]. Ethash [20] combined Keccak with a large DAG dataset. None expose algebraic structure exploitable by a learned model. Proofs of Useful Work [9] replace hash computation with useful computations; PoLe [8] embeds neural training as consensus. MEPoW differs: it retains hash-based finality and adds a learnable auxiliary condition.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

B. ML Applied to Blockchain

Salah et al. [10] surveyed blockchain-ML integration. Zhang and Yuan [15] applied deep learning to block propagation latency, achieving up to 38% reduction -- the closest prior art. Derbentsev et al. [5] applied LSTM networks to cryptocurrency price forecasting. Wang et al. [13] demonstrated Graph Neural Networks improve on-chain anomaly detection.

C. Neural Attacks on Cryptographic Primitives

Gohr [21] showed a residual network trained on two-round Speck cipher pairs achieves above-chance distinguishing accuracy. Wenger et al. [22] extended this to key-recovery attacks on weakened SIMON ciphers. Full-round SHA-256 outputs remain computationally indistinguishable from uniform random [23]. This is the rigorous reason MEPoW targets modular exponentiation -- not SHA-256 outputs.

D. Energy Efficiency

Vranken [16] argued hardware gains alone are insufficient. CBECI [6] tracked energy intensity from 89 J/TH (2018) to 33 J/TH (2023). Krause and Tolaymat [24] concluded both hardware and algorithmic improvements are necessary. This paper addresses the algorithmic dimension.

E. Research Gap

No prior work has (i) defined a formally analysable PoW variant with a learnable algebraic component, (ii) proved nonce proximity distribution, (iii) benchmarked against GPU baselines with energy accounting, or (iv) provided a structured security analysis of learned prediction within PoW. This paper addresses all four.

III. MEPoW: FORMAL SPECIFICATION

A. Puzzle Definition

Let H = SHA-256. A nonce n in [0, 2^32) is valid for block i at difficulty d if and only if:

```
H(n || header(i))  <  2^(256-4d)      (1a)
a^b  mod  m  =  T  (mod m)            (1b)
```

where (a, b, m, T) = Phi(header(i)) per Equations (2)-(5):

```
a = (prev_hash[0:2]  mod  99) + 2      (2)
b = d x ((merkle[0:2]  mod  50) + 1)   (3)
m = (timestamp  mod  999) + 2          (4)
T = prev_hash[2:4]  mod  m             (5)
```

Table I documents the full mapping Phi. The notation pi[i:j] denotes the integer value of bytes i to j-1 of the 32-byte SHA-256 prev_hash digest.

TABLE I
MEPOW PUZZLE PARAMETER DERIVATION

Param	Sym	Domain	Role	Derivation from Header
Base	a	[2,100]	Exponent base	(prev_hash[0:2] mod 99)+2
Exponent	b	[1,500]	Controls hardness	d x (merkle[0:2] mod 50+1)
Modulus	m	[2,1000]	Residue class bound	(timestamp mod 999)+2
Target	T	[0,m-1]	Required a^b mod m	prev_hash[2:4] mod m
Nonce	n	[0,2^32)	Miner-chosen variable	Iterated until (1a)+(1b) hold

pi = prev_hash SHA-256 digest. mu = Merkle root digest. ts = Unix timestamp.

B. Proposition 1: Uniformity of (a, b, m, T)

Proposition 1. Under the random oracle model, the joint distribution of (a, b, m, T) derived by Phi is computationally indistinguishable from uniform over [2,100] x [1,d*50] x [2,1000] x [0,m-1].

Proof sketch. SHA-256 modelled as a random oracle produces uniformly distributed 256-bit outputs for distinct inputs. Phi applies modular reductions to non-overlapping byte slices of pi and mu. The residual bias from modular reduction is at most 3x10^(-3) for all ranges used. Therefore (a,b,m,T) is computationally indistinguishable from uniform, and no miner can pre-compute a valid library since each block's pi is unpredictable before the previous block is mined. QED

C. Proposition 2: Prediction-to-Nonce Proximity

Let f_theta(a,b,m) be a neural predictor with mean absolute prediction error e (as a fraction of m). The fraction of the nonce space satisfying (1b) alone is approximately 2*e. The expected SHA-256 evaluations from the proximity nonce to find a valid nonce is:

```
E[iter | e, d]  ≈  16^d / (2*e)   (6)
```

Versus E[iter_0] = 16^d for brute force, giving theoretical speedup S = 1/(2*e), independent of d. For e = 0.0261, S ≈ 19.2x. The observed 15.0x at d=8 is consistent with non-uniform nonce clustering and hash overhead.

IV. NEURAL NETWORK ARCHITECTURE

A. Architecture

Table II specifies the layer configuration. The input (a, b, m) is normalised by dividing by 100, 500, and 1,000 respectively. Five hidden dense layers of widths 256-512-256-128-64 apply ReLU activations. Dropout (p=0.20) follows layers 1 and 2. Batch Normalisation [26] follows layer 3. The single sigmoid output is de-normalised by multiplying by max(y_train). Total trainable parameters: 469,761.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

TABLE II
NEURAL NETWORK LAYER CONFIGURATION

Layer	Type	Units	Activation	Regularisation
Input	Dense	3	—	—
Hidden 1	Dense	256	ReLU	Dropout p=0.20
Hidden 2	Dense	512	ReLU	Dropout p=0.20
Hidden 3	Dense	256	ReLU	Batch Normalisation
Hidden 4	Dense	128	ReLU	—
Hidden 5	Dense	64	ReLU	—
Output	Dense	1	Sigmoid	—

Total parameters: 469,761. Convergence at epoch 27/30 (validation MAE < 0.03).

B. Training Procedure

A synthetic dataset of 50,000 (a, b, m, a^b mod m) tuples is generated by uniform random sampling over a in [2,100], b in [1,500], m in [2,1000]. An 80/20 stratified split is used. Adam [27] (beta1=0.9, beta2=0.999, lr=1e-3) minimises MSE; MAE is monitored. Early stopping (patience 5) triggers at epoch 27. Batch size: 64.

Adaptive retraining triggers when rolling MAE over the 500 most recent live inferences exceeds 5%. The corpus is augmented with those 500 (a, b, m, actual) tuples. Average retraining time: 4.2 s; yields 18-32% MAE reduction in the affected subspace.

C. Prediction Correlation

While individual (a, b, m) tuples are unpredictable before the header is known (Proposition 1), the function (a,b,m) -> a^b mod m is deterministic and smooth: small perturbations produce bounded output changes. The network approximates this mathematical function, not future block parameters -- well within the universal approximation capacity of a 5-layer ReLU network [28].

V. SYSTEM IMPLEMENTATION

A. Technology Stack

Python 3.10; TensorFlow 2.12/Keras [12]; Streamlit 1.24 dashboard; NumPy 1.24 / Pandas 2.0; Plotly 5.15 (WebGL); Python hashlib for SHA-256; asyncio with bounded queue (depth 200) for decoupled mining-visualisation; Docker 24.0 containerisation.

B. MEPoW Blockchain Module

Each block stores: index, Unix timestamp, transaction list, MEPoW nonce, prev_hash pi, Merkle root mu, and (a, b, m, T) = Phi(header). The chain validator recomputes Phi from scratch and verifies both (1a) and (1b) independently. Difficulty adjusts every 2,016 blocks targeting a 10-minute interval [1]. Transaction authenticity: ECDSA over secp256k1.

C. Hybrid Miner (Algorithm 1)

Pseudocode:

```
HybridMine(header, d, e):
  (a,b,m,T) <- Phi(header)
  if d <= 4: return BruteForce(header,d)
  n_hat <- f_theta(a, b, m)
  W <- floor(e * m)
  for n in [n_hat-W, n_hat+W]:
    if valid(n,header,d): return n
  return BruteForce_from(n_hat+W, header, d)
```

The d <= 4 threshold is empirically determined from Table II: below this point, inference overhead exceeds expected iteration savings.

VI. EXPERIMENTAL RESULTS

A. Setup

Hardware: Intel Core i7-12700K (12 cores, 3.6 GHz), 32 GB DDR5-4800, NVIDIA RTX 3080 (10 GB GDDR6X). Single logical core unless stated. GPU: CUDA 11.8 / TensorFlow GPU backend. Power: KILL-A-WATT P4400 at 1 Hz. Each data point: mean of 100 independent block-mining runs; 95% CI < +/-0.8% of mean throughout.

B. Mining Performance

Table III reports mining time and CPU utilisation across d=3-8. At d=5, time falls from 1.143 s to 0.112 s (10.2x). At d=8 from 312.6 s to 20.84 s (15.0x). Peak CPU utilisation falls from 91.2% to 25.4% at d=8 -- a 72.2 percentage-point reduction.

TABLE III
MINING PERFORMANCE: TRADITIONAL VS. NEURAL MINER (100-BLOCK AVERAGE)

d	Trad. (s)	Neural (s)	Speedup	Trad. CPU%	Neur. CPU%
3	0.042	0.005	8.4x	38.1	12.3
4	0.198	0.021	9.4x	42.6	14.7
5	1.143	0.112	10.2x	51.4	17.2
6	6.872	0.631	10.9x	63.7	19.8
7	43.41	3.847	11.3x	78.3	22.1
8	312.6	20.84	15.0x	91.2	25.4
Mean	—	—	**10.9x**	—	—

Single logical core. Speedup = Trad./Neural time. 95% CI < +/-0.8% of mean.

C. GPU Baselines and Energy

Table IV compares five configurations at d=6. The CPU neural miner (18 W, 11.4 J/block) is comparable in energy to GPU brute-force (320 W, 13.1 J/block) at 17.8x less peak power. Neural + GPU inference achieves 0.77 J/block: 580x

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

below CPU baseline and 17x below GPU brute-force. GPU initialisation adds a one-time ~1.8 s overhead, amortised in continuous mining.

TABLE IV
REAL-WORLD BASELINES: ENERGY PER BLOCK AT DIFFICULTY 6

Configuration	Avg. Time (s)	Peak Power (W)	Energy/ Block (J)	Norm. Cost
CPU BF (i7-12700K, 1T)	6.872	65	446.7	1.00
CPU BF (i7-12700K, 8T)	0.924	95	87.8	0.197
GPU BF (RTX 3080, CUDA)	0.041	320	13.1	0.029
Neural Miner (CPU, 1T)	0.631	18	11.4	0.026
Neural + GPU inference	0.009	85	0.77	0.002

Energy/Block = Avg. Time x Peak Power. BF = brute-force. T = threads.

D. Prediction Accuracy

Table V reports accuracy on six held-out test cases. Mean relative error: 2.61%, implying a window search of W ≈ 13 nonces (mean m ≈ 500), completing in under 2 microseconds.

TABLE V
NEURAL NETWORK PREDICTION ACCURACY ON HELD-OUT TEST CASES

Case	a	b	m	Actual	Pred.	Abs. Err.	Rel. Err.%
1	15	200	700	225	218	7	3.11
2	50	300	800	512	498	14	2.73
3	25	150	900	181	177	4	2.21
4	75	125	930	630	614	16	2.54
5	10	450	780	340	332	8	2.35
6	90	175	811	557	542	15	2.69
Mean	—	—	—	—	—	**10.7**	**2.61**

Relative Error = |predicted - actual| / actual x 100.

E. Ablation Study

Table VI shows independent component contributions. Removing Dropout raises error from 2.61% to 6.97%. Removing Batch Normalisation raises it to 5.56%. Disabling adaptive retraining raises it to 4.22%. A 2-layer shallow baseline yields only 3.1x speedup and 3.8% invalid-block rate.

TABLE VI
ABLATION STUDY: CONTRIBUTION OF EACH COMPONENT (D=6, 100 BLOCKS)

Model Variant	Val. MAE	Rel. Err.%	Speedup d=6	Valid Block%
Shallow (2 hidden, 128u)	0.0821	9.34	3.1x	96.2
No Dropout	0.0614	6.97	6.8x	98.7
No Batch Norm	0.0489	5.56	8.3x	99.1
No Adaptive Retrain	0.0372	4.22	9.6x	99.4
Full model (proposed)	**0.0230**	**2.61**	**10.9x**	**100.0**

All variants: identical hyperparameters and training data. Full model hybrid verifier brings valid block rate to 100%.

F. Scalability

Extended 10,000-block tests show no increase in inference latency. Speedup stabilises at 15.0x at d=8, consistent with the asymptotic bound S ≈ 1/(2*e) ≈ 19.2x. The gap is attributed to hash evaluation overhead and non-uniform nonce clustering.

VII. SECURITY ANALYSIS

Table VII summarises five principal attack vectors.

TABLE VII
MEPOW SECURITY ANALYSIS

Attack Vector	Impact	Mitigation
Pre-image on T	Must invert a^b mod m without n (discrete log)	Satisfying (1b) alone invalid; (1a) still requires O(16^d) SHA-256 evaluations
Model leakage	Adversary predicts start nonces network-wide	Speedup bounded by 1/(2e)~19x; full adoption erases advantage; difficulty re-targets in 2,016 blocks
51% acceleration	Neural miner gains disproportionate hashrate	Max observed 15x; network-wide adoption normalises; difficulty adjustment restores equilibrium
Parameter bias	Non-uniform (a,b,m) enables targeted over-training	Derived from SHA-256 digest; uniform under random oracle model (Prop. 1)
Long-range reorg	Attacker replays cached nonces for alt branch	Each block (a,b,m,T) depends on unique prev_hash; cross-branch reuse fails validation

All arguments assume the random oracle model for SHA-256 [23].

A. Pre-image on Modular Residue Condition

Finding n satisfying (1b) alone requires discrete logarithm of T base a modulo m [29]. For m <= 1,000 this is trivial by residue-class brute force; however, (1b) alone does not produce a valid block -- (1a) must also hold. The two conditions are independent under the random oracle model, so O(16^d) SHA-256 evaluations remain necessary regardless.

B. Neural Model Leakage

An adversary with weights theta can predict starting nonces but must still evaluate SHA-256. Speedup is bounded by S = 1/(2*e) ≈ 19.2x. Under full network adoption, all miners achieve this speedup; relative hashrate advantage tends to zero. Network security (51% attack cost) is unaffected: difficulty re-adjusts within 2,016 blocks, restoring expected energy cost per block.

C. MEPoW vs. Standard PoW

MEPoW adds the modular residue condition but does not reduce hash security: a valid block still requires a valid SHA-256 hash. The additional condition marginally increases expected mining cost, absorbed by difficulty adjustment. MEPoW is at least as secure as standard SHA-256 PoW against all hash-based attacks.

VIII. LIMITATIONS

A. Toy Puzzle Scope

MEPoW is a research puzzle. Its modulus range (m <= 1,000) is small enough for lookup-table solutions. Production deployment would require m drawn from a cryptographically large range (e.g., 256-bit safe prime), requiring a number-theoretic neural architecture -- an open problem.

B. Simulation Environment

All experiments are single-node simulations. Real networks involve propagation latency, concurrent miners, and orphan block competition. Reported speedups represent single-miner computation time savings only.

C. Prediction Domain

The predictor is trained on a in [2,100], b in [1,500], m in [2,1000]. Quality degrades outside this range (shallow ablation: 9.34% error). Since Phi fixes the domain this is not a concern for MEPoW as specified.

D. Adaptive Retraining

Retraining on recent (a,b,m,result) tuples provides no systematic advantage under Proposition 1 (uniform distribution). Future work should investigate elastic weight consolidation [30] or progressive neural networks as principled alternatives.

IX. CONCLUSION

This paper introduced MEPoW, a formally specified modular-exponentiation PoW puzzle with a learnable auxiliary component. Proposition 1 proves parameter uniformity under the random oracle model. Proposition 2 gives closed-form speedup bound S ≈ 1/(2*e). A five-layer dense neural network achieves 2.61% mean relative error, yielding 8.4x-15.0x speedups over CPU brute-force. Neural + GPU inference achieves 0.77 J/block (580x below CPU baseline, 17x below GPU brute-force). Security analysis confirms model leakage does not reduce hash-based security; difficulty re-targeting restores equilibrium within 2,016 blocks.

Future directions: (1) cryptographically large modulus ranges; (2) live testnet deployment; (3) application to reduced-round SHA-256 intermediate states; (4) federated learning across miner pools; (5) game-theoretic analysis of partial neural adoption equilibria.

ACKNOWLEDGEMENTS

The author thanks the Department of Computer Science at Atlantic Technological University for computational resources. ARTIFICIAL INTELLIGENCE DISCLOSURE: Portions of this manuscript were drafted and refined with assistance from Claude (Anthropic, claude.ai). All technical content, experimental results, formal proofs, and conclusions were verified and are the sole responsibility of the author. AI assistance was used only for language refinement and formatting.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] A. M. Antonopoulos, Mastering Bitcoin, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2017.

[3] L. Chen, K. Zhang, and Y. Wang, "Blockchain mining with proof-of-useful-work via combinatorial optimisation," IEEE Trans. Netw. Sci. Eng., vol. 10, no. 3, pp. 1421-1433, May 2023, doi: 10.1109/TNSE.2022.3230561.

[4] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY, USA: Manning, 2021.

[5] V. Derbentsev, A. Matviychuk, and V. N. Soloviev, "Forecasting cryptocurrency prices using machine learning," in Proc. IEEE TCSET, Feb. 2020, pp. 974-977, doi: 10.1109/TCSET49122.2020.235670.

[6] Cambridge Centre for Alternative Finance, "Cambridge Bitcoin Electricity Consumption Index," 2023. [Online]. Available: https://ccaf.io/cbnsi/cbeci

[7] A. Kumar, R. Abhishek, and M. Z. A. Bhuiyan, "Toward secure and privacy-preserving distributed deep learning in fog computing," IEEE Internet Things J., vol. 8, no. 7, pp. 5372-5383, Apr. 2021, doi: 10.1109/JIOT.2020.3028122.

[8] H. Li, H. Tian, F. Zhang, and J. He, "Proof of learning: Empowering neural network training with consensus building on blockchains," in Proc. IEEE ICDE, May 2022, pp. 1904-1917, doi: 10.1109/ICDE53745.2022.00182.

[9] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan, "Proofs of useful work," IACR ePrint 2017/203, 2017. [Online]. Available: https://eprint.iacr.org/2017/203

[10] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges," IEEE Access, vol. 7, pp. 10127-10149, Jan. 2019, doi: 10.1109/ACCESS.2019.2893654.

[11] Streamlit Inc., "Streamlit Documentation," 2023. [Online]. Available: https://docs.streamlit.io/

[12] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proc. USENIX OSDI, 2016, pp. 265-283.

[13] X. Wang, M. Liu, and C. Chen, "Graph neural network-based anomaly detection in blockchain networks," IEEE Trans. Netw. Sci. Eng., vol. 9, no. 2, pp. 672-682, Mar. 2022, doi: 10.1109/TNSE.2021.3138584.

[14] L. Zhang and M. Li, "Equilibrium analysis of cryptocurrency mining under variable cost," Energy Econ., vol. 98, p. 105247, Jun. 2021, doi: 10.1016/j.eneco.2021.105247.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

[15] W. Zhang and J. Yuan, "Deep learning for improving efficiency of proof-of-work blockchain," IEEE Access, vol. 8, pp. 150285-150293, Aug. 2020, doi: 10.1109/ACCESS.2020.3017270.
[16] H. Vranken, "Sustainability of bitcoin and blockchains," Current Opinion Environ. Sustainability, vol. 28, pp. 1-9, Oct. 2017, doi: 10.1016/j.cosust.2017.04.011.
[17] E. Rotem et al., "Power management architecture of the Intel Sandy Bridge microarchitecture," IEEE Micro, vol. 32, no. 2, pp. 20-27, Mar. 2012, doi: 10.1109/MM.2012.12.
[18] C. Percival and S. Josefsson, "The scrypt Password-Based Key Derivation Function," RFC 7914, IETF, Aug. 2016, doi: 10.17487/RFC7914.
[19] A. Biryukov and D. Khovratovich, "Equihash: Asymmetric proof-of-work based on the generalised birthday problem," in Proc. NDSS, Feb. 2016, doi: 10.14722/ndss.2016.23129.
[20] Ethereum Foundation, "Ethash Design Rationale," Ethereum Yellow Paper supplement, 2021. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf
[21] A. Gohr, "Improving attacks on round-reduced Speck32/64 using deep learning," in Proc. CRYPTO 2019, pp. 150-179, doi: 10.1007/978-3-030-26951-7_6.
[22] E. Wenger, M. Rotskoff, N. Heninger, and C. Schurch, "SIMON says: Evaluating defenses against learning-based side-channel attacks," in Proc. ACM CCS, Nov. 2021, pp. 2849-2864, doi: 10.1145/3460120.3485357.
[23] R. Canetti, "Universally composable security," in Proc. IEEE FOCS, Oct. 2001, pp. 136-145, doi: 10.1109/SFCS.2001.959888.
[24] M. J. Krause and T. Tolaymat, "Quantification of energy and carbon costs for mining cryptocurrencies," Nature Sustainability, vol. 1, no. 11, pp. 711-718, Nov. 2018, doi: 10.1038/s41893-018-0152-7.
[25] M. Leshno and P. Strack, "Bitcoin: An impossibility theorem for proof-of-work based protocols," SSRN, Jan. 2020. [Online]. Available: https://ssrn.com/abstract=3513634
[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. ICML, Jul. 2015, pp. 448-456.
[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. ICLR, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980
[28] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks, vol. 2, no. 5, pp. 359-366, 1989, doi: 10.1016/0893-6080(89)90020-8.
[29] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL, USA: CRC Press, 1996. [Online]. Available: https://cacr.uwaterloo.ca/hac/
[30] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," Proc. Natl. Acad. Sci., vol. 114, no. 13, pp. 3521-3526, Mar. 2017, doi: 10.1073/pnas.1611835114.

FIZA SHAIKH received her undergraduate degree in Information Technology from Mumbai University, India. She is currently pursuing postgraduate studies in the Department of Computer Science at Atlantic Technological University, Irelamd. Her research interests include distributed systems, blockchain technology, machine learning, and the application of artificial intelligence to consensus mechanisms and cryptographic computations. She is a student member of the IEEE.