FHEMaLe: Framework for Homomorphic Encrypted Machine Learning

B. Pradeep Kumar Reddy[1], Sameeksha Goyal[2], Ruchika Meel[3], Ayantika Chatterjee[4]

pradeepkumarreddy.bukka@gmail.com, sameeksha.goyal.810@gmail.com, ruchikasingh758@gmail.com, cayantika@gmail.com

Indian Institute of Technology, Kharagpur, India-721302

Aim: The FHEMaLe framework enables privacypreserving machine learning (ML) on both edge and cloud environments using Fully Homomorphic Encryption (FHE). It addresses the privacy challenges of edge ML, where resource-constrained devices process sensitive data, and cloud ML, where data is outsourced to honest-but-curious servers. By performing computations on encrypted data, FHEMaLe ensures confidentiality while supporting scalable ML on diverse platforms.

Motivation

- **Edge and Cloud Synergy:** Edge ML reduces latency and bandwidth needs, while cloud ML supports large-scale data processing. Both require robust privacy measures.
- **Privacy Challenges:** Edge devices face physical capture risks, and cloud servers are vulnerable to data breaches in honest-but-curious models.
- **FHE Trade-offs:** CKKS excels in approximate arithmetic for neural networks (NNs) but struggles with exact operations (e.g., comparisons in KNN, SVM). TFHE supports bit-level logic but is computationally intensive.

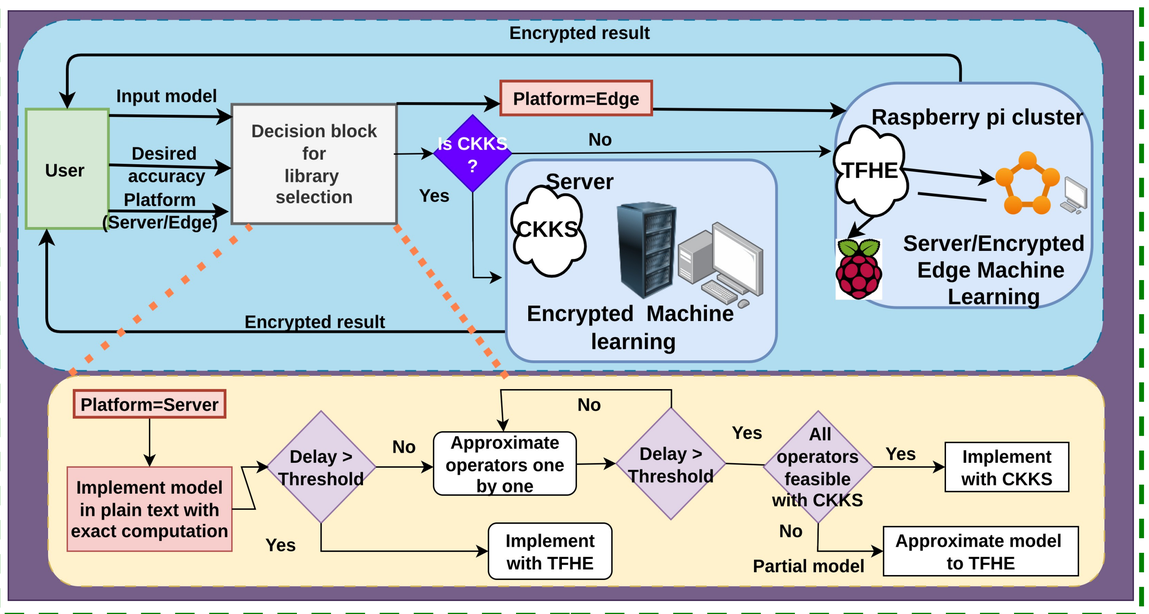

Fig. 1: Encrypted Machine Learnig as a Service

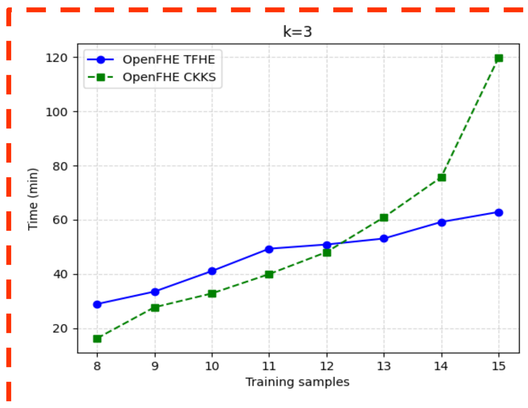

Fig. 2: CKKS vs TFHE KNN computaion time on cloud

Why KNN is Not Suitable in Encrypted Domain?

- KNN requires exact comparisons and sorting for distance-based classification, which are challenging in FHE.
- CKKS enables homomorphic distance computations for KNN efficiently, secure and efficient neighbor selection (sorting of encrypted data) scales poorly remains a critical challenge.

Methodology

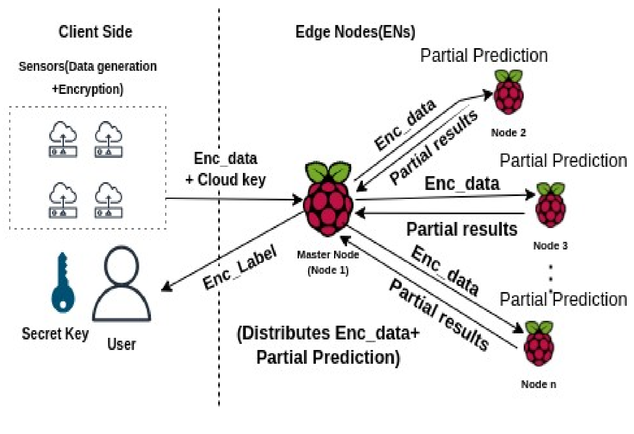

Fig. 3: Distributed Framework.

FHEMaLe evaluates ML models in plaintext to select CKKS or TFHE based on required operators and platform (edge or cloud). Key components include:

- **Library Selection:** CKKS for SIMD-based operations (e.g., matrix multiplications in NNs) in cloud environments; TFHE for exact comparisons (e.g., KNN, SVM) on edge devices.
- Distributed processing for edge (Raspberry Pi clusters) and cloud (highresource servers) to optimize performance.
- **Distributed Processing:** Edge nodes compute partial decision functions (PDFs), aggregated by a master node (Node1). Cloud servers handle largescale encrypted computations.
- **Encrypted Operations:** FHE circuits (e.g., FHE_Adder, FHE_Mux) support secure dot products and label predictions.
- Novel encrypted approximate division algorithm for efficient sigmoid computation in logistic regression (LR).
- Evaluation and minimization of computational overhead introduced by encrypted data operations with distributed and concurrent computing on the edge devices network is also explored.

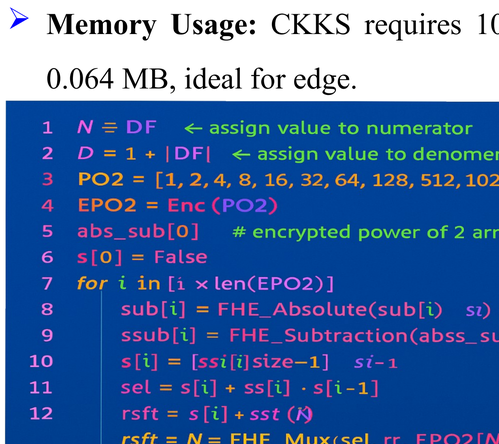

Fig 4: Encrypted Distributed LR Framework.

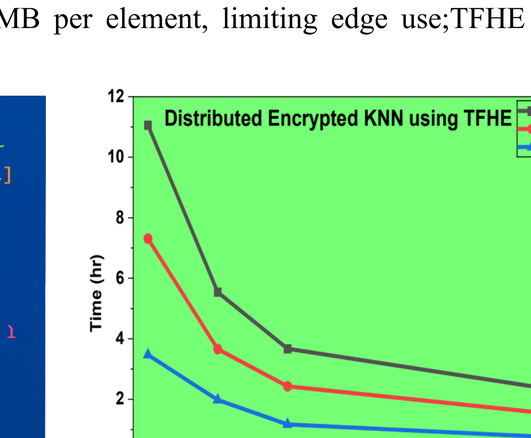

Fig 5: Encrypted Distributed SVM Framework.

Why CKKS is Suitable for Neural Networks?

CKKS is ideal for NNs due to its support for approximate arithmetic and SIMD operations:

- **Vectorized Operations:** CKKS efficiently handles matrix multiplications and dot products, core to NN layers (e.g., linear, attention heads).
- **SIMD Packing:** Enables parallel evaluation of multiple slots, improving throughput for high-dimensional embeddings.
- **Tolerable Approximations:** NNs tolerate minor inaccuracies in operations like ReLU or sigmoid, unlike KNNs need for exact comparisons.
- **Cloud Efficiency:** CKKS achieves faster NN processing on high-resource servers.

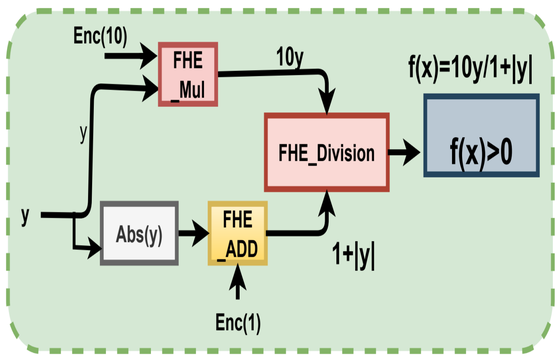

Fig 6: Fastsigmoid with Encrypted Gates.

Results and discussions

- **Cloud Performance:** On an HP Z240 (Intel Xeon, 64 GB RAM), CKKS achieves faster KNN for small datasets (248s vs. 780s for TFHE, 4 samples) but slows for larger datasets (7182s vs. 3768s, 15 samples).
- **Edge Performance:** Distributed processing on 11 Raspberry Pi nodes reduces KNN time (37 min), SVM (4.15 min), and LR (7.82 min) using OpenFHE.
- **Memory Usage:** CKKS requires 107 MB per element, limiting edge use;TFHE uses 0.064 MB, ideal for edge.

```
1  N ≡ DF    ← assign value to numerator
2  D = 1 + |DF|   ← assign value to denomerator
3  PO2 = [1, 2, 4, 8, 16, 32, 64, 128, 512, 1024,...]
4  EPO2 = Enc (PO2)
5  abs_sub[0]    # encrypted power of 2 array
6  s[0] = False
7  for i in [1 × len(EPO2)]
8      sub[i] = FHE_Absolute(sub[i])   si
9      ssub[i] = FHE_Subtraction(abss_sub[−1
10     s[i] = [ssi[i]size−1]   si−1
11     sel = s[i] + ss[i] · s[i−1]
12     rsft = s[i] + sst (K)
       rsft = N = FHE_Mux(sel, rr, EPO2[N)
15     # Shifts if D > EPO2[i] or if
       D < EPO2[i] but closer than EPO2[i−1]
16 ENDFOR
```

Algo 1: Encrypted Approximate Division

Fig. 7: Encrypted KNN Prediction Time

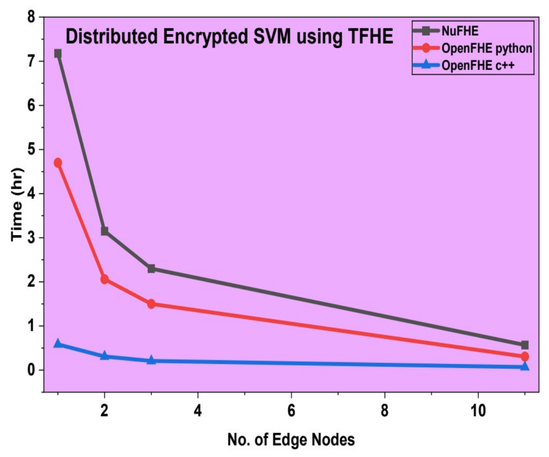

Fig. 8: Encrypted SVM Prediction Time

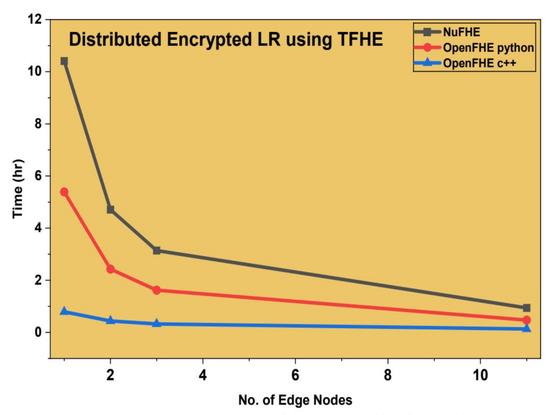

Fig. 9: Encrypted LR Prediction Time

Table 1: Ckks vs TFHE Time (single prediction) and Memory per element (one element 16 bit) on cloud

ML Model	OpenFHE CKKS Time (sec)	OpenFHE TFHE Time (sec)	OpenFHE TFHE Memory (MB)	OpenFHE CKKS Memory (MB)
KNN	7182	3768	0.064	107
SVM	18	946	0.064	107
LR	17	650	0.064	107
NN	57	3982	0.064	107

Future Directions

- **Optimize FHE circuits for additional ML algorithms (e.g., Decision Trees).**
- **Scale cloud and edge clusters for larger datasets and complex models.**
- **Develop hybrid CKKS-TFHE approaches for balanced performance.**
- **Explore energy-efficient FHE for low-power edge devices.**

References

1. Chatterjee, A., & Sengupta, I. (2015). Translating algorithms to handle fully homomorphic encrypted data on the cloud. *IEEE Transactions on Cloud Computing*, *6*(1), 287-300.
2. **Reddy, B. P. K.**, Meel, R., & Chatterjee, A. (2024). Encrypted KNN Implementation on Distributed Edge Device Network. *IACR Cryptol. ePrint Arch.*, *2024*, 648.
3. Agrawal, R., & Joshi, A. (2023). On architecting fully homomorphic encryption-based computing systems. Springer International Publishing.
4. **Pradeep Kumar Reddy, B.,** & Chatterjee, A. (2025). SMLaaS: Secure Machine Learning as a Service Ensuring Data and Model Parameter Privacy. *Security and Privacy*, *8*(4), e70047.

Connect With Us