# Privacy-Preserving Financial Fraud Detection Through Secure Neural Network Inference and Adaptive Polynomial Activation Optimization

Eoin Bohan
Department of Computer Science, University of Central Florida, Orlando, FL, USA.
eoinmail@ucf.edu

Kristopher Orr
Department of Computer Science, George Mason University, Fairfax, VA, USA.
kristopher.work@gmu.edu

**Abstract**

Financial fraud detection increasingly depends on neural models that consume detailed transaction records, account histories, and behavioral signals. These data are confidential, and practical deployment therefore requires inference that protects both customer features and proprietary model parameters. Secure neural network inference using homomorphic encryption or multiparty computation offers a principled cryptographic route, but nonlinear activation functions dominate the computational cost and constrain permissible network depth. Fixed low-degree activation replacements can reduce this cost, yet they are usually selected without explicit attention to the cost-sensitive decision boundary of fraud detection or to the multiplicative-depth budget of the secure protocol. This article develops an adaptive polynomial activation optimization framework for privacy-preserving financial fraud detection. The framework treats activation degree, polynomial coefficients, and decision threshold as coupled design variables. It formulates fraud detection as a cost-sensitive risk problem, derives a Chebyshev polynomial family for encrypted evaluation, and optimizes activation approximations under multiplicative-depth and latency constraints. A calibration analysis connects polynomial approximation error to threshold selection and expected fraud cost, while an error-propagation bound characterizes how activation error accumulates across layers. The main insight is that privacy-preserving fraud detection should not minimize activation approximation error alone; it should allocate polynomial capacity according to its effect on fraud cost, score calibration, and secure computation depth. The resulting methodological framework provides a defensible basis for designing secure neural inference systems whose accuracy, privacy, and operational cost are treated as jointly engineered rather than sequentially optimized.

**Keywords**

privacy-preserving machine learning; secure neural network inference; financial fraud detection; polynomial activation approximation; homomorphic encryption; cost-sensitive calibration

## 1. Introduction

Financial fraud detection is a sequential decision problem conducted under adversarial pressure, severe class imbalance, delayed label availability, and strong confidentiality requirements. Card-not-present fraud, account takeover, money laundering, and payment

authorization all depend on transaction attributes, device signals, merchant context, and historical behavior. Surveys of fraud detection systems emphasize that effective detection requires both predictive modeling and careful integration with operational decision rules [1]. Practitioner studies of credit card fraud illustrate that model performance can degrade when temporal drift, class imbalance, and alert-volume constraints are not reflected in evaluation [2]. Data-mining surveys further show that fraud detection is not a single classification task but a family of problems in which the cost of missed fraud, the cost of false alerts, and the latency of decisions differ across institutions and channels [3].

Neural networks are attractive in this setting because they can represent nonlinear interactions among transaction features, behavioral histories, and contextual variables. Deep learning provides general function-approximation tools for such interactions [4], and residual-style architectures demonstrate how deeper models can improve representation learning when optimization is carefully managed [5]. In financial fraud detection, however, the most sensitive obstacle is not only predictive capacity. Transaction records contain personal and commercially sensitive information, and model parameters may encode valuable institutional knowledge. A fraud detector that requires plaintext transmission of customer data to a third-party model provider creates confidentiality risks that are difficult to justify in many deployment settings.

Cryptographic inference provides a technical alternative. Homomorphic encryption and secure multiparty computation allow computations to be performed over encrypted or secret-shared data, with the goal of revealing only the final result to authorized parties [6]. These techniques have advanced considerably, but they impose a distinctive cost structure. Linear operations such as matrix multiplication can be batched and parallelized, whereas nonlinear activation functions require either deep multiplicative circuits, lookup tables, approximations, or interaction between parties. The result is a privacy-accuracy-latency frontier in which activation design is not a minor implementation detail but a central modeling decision.

This article addresses that decision for financial fraud detection. The central contribution is an adaptive polynomial activation optimization framework for secure neural network inference. The framework formulates fraud detection as a cost-sensitive risk problem, represents activation functions by degree-controlled Chebyshev polynomials, and optimizes polynomial degree and coefficients under a multiplicative-depth budget. It further couples the approximation with threshold calibration, because a polynomial substitute that preserves average score accuracy can still shift the decision boundary in ways that alter false-positive and false-negative costs. The approach is methodological and analytical: it derives design conditions, objective functions, and error-propagation relationships that are independent of any particular proprietary transaction corpus. The manuscript thereby treats secure inference as a co-design problem among cryptographic cost, fraud-specific decision loss, and calibration stability.

The remainder of the article is organized as follows. Section 2 defines the fraud detection problem and the secure inference threat model. Section 3 examines the computational bottlenecks of secure neural inference and explains why activation replacement is the critical design variable. Section 4 develops the adaptive polynomial activation optimization framework. Section 5 analyzes fraud-specific calibration, threshold selection, and error propagation. Section 6 discusses implementation logic, sensitivity, and failure modes. Section 7 situates the framework relative to prior secure inference work and states boundary conditions. Section 8 concludes.

## 2. Problem Formulation and Threat Model

### 2.1. Cost-sensitive fraud inference

Let x ∈ R^m denote a transaction-level feature vector containing amount, time, merchant category, device attributes, prior account activity, and other risk indicators. Let y ∈ {0,1} denote the fraud label, with y = 1 indicating fraud. A neural fraud scorer with L layers produces a risk score through alternating linear and nonlinear transformations. For layer l, the pre-activation and activation are written as follows.

Here $a^{(0)} = x$ is the input feature vector, $W^{(l)}$ is the weight matrix of layer l, $b^{(l)}$ is the bias vector, φ is a nonlinear activation function, and $a^{(l)}$ is the layer output. The final score s(x) is a scalar function of $a^{(L)}$, typically obtained by a linear output layer followed by a logistic or monotone transformation. The model can be trained with any suitable fraud-detection loss, but deployment requires a decision rule that maps the score to an action. In practice, the action may be approval, review, step-up authentication, or block.

Fraud detection is cost-sensitive because false negatives and false positives have asymmetric consequences. A missed fraud may produce direct financial loss, remediation cost, and regulatory exposure. A false positive may inconvenience customers, increase manual review workload, and damage legitimate commerce. Let $C^-$ denote the cost of a false negative and $C^+$ denote the cost of a false positive. For a threshold τ, the empirical decision cost over a labeled sample is expressed in the next equation.

$$C(\tau) = C^- \sum\nolimits_{-}\{y = 1\}1\{s(x) < \tau\} + C^+ \sum\nolimits_{-}\{y = 0\}1\{s(x) \geq \tau\}$$

In this expression, $1\{\cdot\}$ is the indicator function, the first summation aggregates missed fraud cases whose scores fall below the threshold, and the second aggregates legitimate transactions whose scores meet or exceed the threshold. The costs $C^-$ and $C^+$ are institution-specific and may vary by transaction value or customer segment. The equation makes explicit that threshold selection is part of the fraud model, not an external afterthought. A secure inference design that preserves average score accuracy but degrades calibration near τ can therefore increase operational cost even when its aggregate classification error appears acceptable.

### 2.2. Secure inference model

The secure inference setting considered here involves two logical parties. The client, such as a payment processor or financial institution, holds transaction features x. The model provider holds the trained neural network parameters $W^{(l)}$ and $b^{(l)}$. The parties wish to compute s(x) without revealing x to the model provider or the model parameters to the client beyond the intended output. In a homomorphic encryption deployment, the client encrypts x and the provider evaluates the network over ciphertexts. In a secret-sharing deployment, the parties or a set of servers evaluate the network over shares and reconstruct only the authorized output. Hybrid protocols may combine homomorphic encryption, garbled circuits, and oblivious transfer.

The threat model assumed in this analysis is the standard semihonest or passive model: parties follow the protocol but may attempt to infer protected information from transcripts. This assumption is appropriate for articulating the privacy-accuracy-cost problem, although production systems may require active security, leakage protection, and defenses against malformed inputs. The framework does not rely on differential privacy, because the objective is cryptographic confidentiality of inputs and weights rather than statistical

indistinguishability of released aggregates. The relevant privacy claim is therefore conditional on the cryptographic protocol and its implementation.

### 2.3. Design requirements

Three requirements interact in secure fraud detection. The first is decision quality under asymmetric costs. The secure model must preserve the ranking and calibration needed to separate high-risk transactions from legitimate ones at the operative threshold. The second is cryptographic feasibility. The network must fit within the multiplicative-depth and latency budget of the chosen protocol. The third is calibration stability. Because polynomial activation replacement changes the mapping from pre-activation values to scores, the threshold and any calibration layer must be adjusted or analyzed jointly with the activation approximation.

These requirements imply that activation design should be layer-aware and decision-aware. A uniform replacement of every activation by the same low-degree polynomial may be convenient, but it ignores differences in layer sensitivity, input ranges, and the location of the fraud decision boundary. The adaptive framework developed in Section 4 treats activation degree and coefficients as variables constrained by cryptographic cost and optimized with respect to fraud-specific loss.

## 3. Secure Neural Network Inference Bottlenecks

### 3.1. Homomorphic and multiparty cost structure

Fully homomorphic encryption enables arbitrary computation over encrypted data and provides the conceptual foundation for encrypted inference [7]. Leveled homomorphic encryption and related constructions avoid or limit bootstrapping by designing circuits of bounded multiplicative depth [8]. Approximate-number homomorphic encryption has made fixed-point arithmetic and polynomial evaluation more practical for numerical workloads [9]. Notwithstanding these advances, secure inference remains constrained by ciphertext expansion, noise growth, multiplication depth, rotation cost, and communication volume.

Early encrypted neural network systems demonstrated that inference could be performed over encrypted inputs with batching and polynomial approximations [10]. Low-latency frameworks reduced communication and computation through optimized packing, oblivious transfer, and protocol composition [11]. Cryptographic inference services for neural networks further improved practical performance by combining homomorphic encryption with secret sharing and carefully chosen nonlinear protocols [12]. Acceleration techniques for encrypted prediction have also explored polynomial approximations and protocol-level trade-offs [13]. Across these systems, the same structural pattern appears: linear layers scale with matrix dimensions, while nonlinearities determine depth, interaction rounds, and often the dominant latency.

### 3.2. Activation replacement as the critical choice

The rectified linear unit and its variants are standard in deep networks because they avoid saturation and support efficient gradient-based training [16]. Under encryption, however, the rectified linear unit is problematic. It requires a comparison or branching operation, neither of which is native to homomorphic encryption. Secure multiparty computation can evaluate comparisons, but each comparison consumes communication rounds and may break vectorized processing. Replacing the activation with a polynomial removes branching and allows evaluation by additions and multiplications, but the replacement must be chosen

carefully. Automated replacement strategies have been developed to select lower-cost substitutes for rectified linear units in private network inference [14]. Polynomial approximations have also been used to adapt deep networks to encrypted data [15]. These approaches establish the feasibility of activation substitution, but they leave open the fraud-specific question of how much polynomial degree should be assigned to each layer and how the replacement should be calibrated to an asymmetric decision loss.

The required reference is used only to support the general point that automatic activation replacement is a relevant secure-inference technique. The present framework does not depend on the specific replacement policy of that work. Instead, it formulates a broader optimization problem in which polynomial activation degree, coefficients, and fraud threshold are selected jointly under a multiplicative-depth budget. Secure multiparty learning systems provide another route to privacy-preserving training and inference [17], and classification over encrypted data has been studied as a distinct cryptographic workload [18]. Sparsity-aware encrypted inference [19] and fast encrypted deep networks [20] further illustrate that performance depends on the interaction between network architecture, numerical encoding, and nonlinear operations.

## 4. Adaptive Polynomial Activation Optimization

### 4.1. Polynomial family and approximation interval

The framework replaces the activation φ on a bounded interval by a degree-controlled polynomial. Let z denote a pre-activation value and suppose that, after range analysis or clipping, z lies in [−B, B]. A Chebyshev basis is used because it provides stable numerical behavior and a natural way to control approximation error on an interval. The polynomial approximation of degree d is written as follows.

$$p(d,z) = \sum_{k=0}^{d} \kappa_k T_k\left(\frac{z}{B}\right), z \in [-B, B]$$

Here d is the polynomial degree, $\kappa_k$ are coefficients, $T_k$ is the Chebyshev polynomial of degree k, and B is the activation range bound. The argument z/B maps [−B,B] to [−1,1], the standard domain of the Chebyshev basis. The bound B is not cosmetic. If the actual pre-activation distribution extends beyond B, polynomial values can diverge rapidly and corrupt downstream layers. Range control through weight regularization, input clipping, normalization, or per-layer calibration is therefore part of the design.

The polynomial family is flexible enough to approximate sigmoid, hyperbolic tangent, softsign, and smoothed rectified activations. It also permits degree zero, which corresponds to a constant activation and may be appropriate only in layers with little nonlinear contribution. The adaptive problem is to choose d and κ for each layer rather than applying a uniform degree across the network.

### 4.2. Joint optimization objective

The central design objective couples fraud loss, decision cost, and polynomial degree. Let p(d,z;κ) denote the polynomial activation with degree d and coefficient vector κ. Let ℓ be a differentiable surrogate for the fraud classification loss, such as a weighted cross-entropy or a cost-sensitive logistic loss. Let γ weight the decision cost C(τ), and let λ weight the cost of increasing polynomial degree. The adaptive activation problem is stated in the next equation.

$$min\ over\ d, \kappa, \tau\ of\ E[\ell(p(d,z;\kappa),y)] + \gamma C(\tau) + \lambda d, d \le D^{m}$$

In this expression, E[·] denotes expectation over the transaction distribution, C(τ) is the decision cost from Equation 2, τ is the decision threshold, and $D^m$ is the maximum allowed degree for a single activation. The parameter γ controls the emphasis on operational fraud cost versus surrogate classification loss, and λ penalizes degree because higher degree increases multiplication depth and ciphertext noise. The formulation is intentionally general: it can be solved as a post-training calibration problem, or it can be embedded in polynomial-aware training so that the network learns weights that are robust to activation approximation.

At the coefficient level, a regularized approximation objective is useful. For a fixed degree d and a distribution of pre-activation values D_z induced by the current network and transaction data, the coefficients can be selected by minimizing approximation error with a complexity penalty.

$$\kappa\bigstar = argmin\ over\ \kappa\ of\ E[(p(d,z;\kappa) - \varphi(z))^2] + \mu\|\kappa\|^2$$

Here κ★ is the selected coefficient vector, φ(z) is the original activation, and μ is a regularization parameter that discourages large coefficients and reduces sensitivity to numerical noise. The distribution D_z is not necessarily the raw input distribution; it is the distribution of pre-activations under the current model and the client population. This distinction matters because activation approximation should be accurate where the network actually operates, not only on arbitrary points of [−B,B].

### 4.3. Depth-aware degree allocation

In homomorphic encryption, multiplication depth is a scarce resource. A polynomial of degree d requires a chain or tree of multiplications, and deeper circuits consume more noise budget and may require larger parameters or bootstrapping. A conservative depth allocation model sums layerwise polynomial costs. Let $d_l$ be the degree assigned to layer l, L the number of layers, and $D^b$ the total multiplicative-depth budget available for activation evaluation. The depth constraint is expressed in the next equation.

$$D_{total} = \sum_{l=1}^{L} \lceil log_2(d_l + 1) \rceil \leq D^b$$

In this expression, $D_{total}$ is a conservative estimate of the total multiplicative depth consumed by activation evaluation, and $\lceil \cdot \rceil$ denotes the ceiling function. The expression is a design abstraction rather than a claim of exact protocol cost; actual depth depends on the polynomial evaluation algorithm, ciphertext packing, and whether squaring is optimized separately. Its purpose is to force degree allocation to be an explicit resource-allocation problem. If one layer receives a high-degree polynomial, other layers or the network depth may need to be reduced to remain within $D^b$.

This constraint suggests a marginal-allocation rule. Degree should be assigned to the layer where an additional unit of polynomial depth produces the largest reduction in fraud-specific decision loss. In a network with early layers that learn broad feature interactions and later layers that calibrate risk, the most valuable polynomial capacity may not lie in the same layer for every institution. The adaptive framework therefore supports layer-specific degrees rather than a single global degree.

### 4.4. Solution procedure and model integration

A practical solution procedure follows an alternating design. First, a feasible degree vector is initialized under the depth budget, for example with low degrees in all layers and additional capacity in layers identified by sensitivity analysis. Second, for the current degree

vector, coefficients are fitted using the regularized Chebyshev objective in Equation 5, with pre-activation samples drawn from the model's operating distribution. Third, the threshold $\tau$ is recalibrated using validation data and the asymmetric cost in Equation 2. Fourth, a degree reallocation step estimates the marginal benefit of increasing or decreasing degree in each layer and updates the vector while preserving the depth constraint. The process can be repeated until the degree vector and threshold stabilize.

The procedure can be integrated in two ways. In post-training approximation, the original network is trained first and only activations are replaced and calibrated. This approach is simpler but may suffer if the network relies heavily on sharp rectified behavior. In polynomial-aware training, the network is trained with the polynomial activation or with a smooth surrogate whose degree and coefficients are updated during training. This approach can produce weights that are more compatible with polynomial evaluation, at the cost of a more complex optimization. In both cases, the fraud decision threshold must be treated as part of the secure model design, because the approximation changes the score distribution.

## 5. Fraud-Specific Calibration and Decision Analysis

### 5.1. Cost-sensitive threshold calibration

If the model output $p(x)$ is a calibrated probability of fraud, the optimal threshold under the asymmetric costs $C^-$ and $C^+$ follows from comparing expected costs. Predicting fraud is preferred when $C^+(1-p) \le C^- p$, which yields the threshold in the next equation.

$$\tau\bigstar = C^+/(C^+ + C^-)$$

Here $\tau\bigstar$ is the cost-minimizing probability threshold, $C^+$ is the false-positive cost, and $C^-$ is the false-negative cost. For example, if a missed fraud is regarded as much more costly than a false alert, $\tau\bigstar$ will be small, because the model should flag transactions with modest fraud probability. The equation also clarifies why activation approximation can be operationally significant: if polynomial replacement shifts scores near $\tau\bigstar$, the effective cost changes even if the ranking of most transactions is preserved.

In practice, scores may not be perfectly calibrated. A calibration layer, isotonic mapping, or piecewise threshold adjustment can be fitted on validation data. In the secure setting, calibration should be performed before encryption or inside the secure computation if the calibration parameters themselves are sensitive. The adaptive objective in Equation 4 allows $\tau$ to be optimized jointly with the polynomial activation, so the approximation is not evaluated only by mean-squared error against the original activation.

### 5.2. Error propagation and decision stability

Activation approximation error does not remain local. It propagates through linear transformations and subsequent activations. Let $e^{(l)}$ denote the difference between the polynomial-activation layer output and the original layer output at layer $l$. Let $\varepsilon^{(l)}$ bound the supremum approximation error of the polynomial activation on the operating range of layer $l$, and let $L_\varphi$ be the Lipschitz constant of the original activation. A conservative propagation bound is given in the next equation.

$$\|e^{(l)}\| \le \varepsilon^{(l)} + L_\varphi \|W^{(l)}\| \|e^{(l-1)}\|$$

In this expression, $\|\cdot\|$ denotes a suitable vector or operator norm, $W^{(l)}$ is the weight matrix at layer $l$, and $e^{(l-1)}$ is the error entering that layer. The first term captures local approximation error introduced by the polynomial; the second captures amplification of earlier error by the

linear layer and the activation's Lipschitz constant. The bound is conservative, but it provides a useful design principle: layers with large weight norms or large approximation error should receive more polynomial capacity or be regularized during training. The final score error is bounded by the output-layer norm times the terminal layer error, so small activation errors in early layers can still matter if subsequent linear maps amplify them.

Decision stability can be examined through a margin argument. A transaction is stable at threshold $\tau$ if its score remains on the same side of $\tau$ under the bounded score perturbation implied by Equation 8. Transactions with scores close to $\tau$ are the most vulnerable to approximation-induced flips. Because fraud operations often focus attention on borderline alerts, calibration should emphasize the score region around $\tau$ rather than the entire distribution uniformly. This observation supports weighted polynomial fitting that assigns greater weight to pre-activation values associated with borderline decisions.

### 5.3. Tail behavior under class imbalance

Fraud detection is dominated by class imbalance. Most transactions are legitimate, and the fraud class may be rare. A polynomial approximation that minimizes average squared error over all transactions can devote most of its capacity to the high-density legitimate region and underrepresent the high-risk tail. The resulting secure model may appear accurate in aggregate while losing sensitivity in the small set of transactions that should trigger review. The adaptive objective should therefore use class weights, cost weights, or tail-focused validation metrics.

A second tail issue concerns the activation range. Extreme transaction features or adversarial inputs can produce pre-activations outside the interval used for polynomial fitting. A polynomial that is well behaved on $[-B,B]$ may grow rapidly outside that interval. Range clipping, bounded normalization, and monitoring of activation magnitudes are therefore necessary. In fraud applications, adversarial manipulation of transaction features is plausible, so input validation and robust preprocessing are security-relevant components of the secure inference pipeline.

## 6. Implementation and Sensitivity Analysis

### 6.1. Deployment workflow

A deployment of the framework can be described without reference to a specific proprietary system. The client encodes transaction features into a numerical vector and encrypts or secret-shares them according to the protocol. The model provider evaluates linear layers using homomorphic addition and multiplication, applies polynomial activations with the layer-specific degree vector, and returns the encrypted or shared score. The client decrypts or reconstructs the score and applies the calibrated threshold. If the model provider is not permitted to learn the threshold, the threshold can be applied by the client after decryption, or it can be incorporated into a secure comparison at the end of the protocol.

The polynomial coefficients, degree vector, and calibration parameters are part of the model. They should be protected consistently with the weights. Packing strategy affects throughput: multiple transactions can be encoded into a single ciphertext, and polynomial evaluation can be vectorized across slots. However, packing constrains the ability to branch by transaction, which reinforces the value of a branch-free polynomial activation. Secure multiparty systems offer complementary trade-offs: they can evaluate comparisons more directly but incur communication overhead [17]. Classification over encrypted data [18] and

sparsity-aware encrypted inference [19] illustrate that protocol-level optimizations can shift the optimal activation design. Fast encrypted deep-network methods [20] further show that architecture and numerical encoding must be considered together.

### 6.2. Sensitivity and failure modes

The first sensitivity dimension is degree versus latency. Increasing polynomial degree can reduce approximation error, but it increases multiplication depth and may trigger larger cryptographic parameters or more communication rounds. The marginal benefit of degree is highest when the layer's pre-activation distribution has strong curvature on the fitting interval and when errors in that layer influence the decision boundary. It is lowest when the layer is nearly linear or when its output has little effect on the final score.

The second dimension is the activation range B. A conservative large B reduces clipping but forces the polynomial to cover a wider interval, which can reduce accuracy near the operating region. A small B improves local accuracy but increases the risk of divergence outside the interval. The framework therefore requires a range-calibration step, using training or validation pre-activations and possibly robust quantiles rather than maximum observed values.

The third dimension is threshold sensitivity. Small changes in $\tau$ can have large effects on alert volume when the score distribution is concentrated near the threshold. Calibration should therefore be evaluated with respect to both fraud cost and operational capacity. If manual review can process only a limited number of alerts, the threshold is constrained by capacity as well as by cost. The adaptive objective can be modified to include a capacity penalty, but the core structure remains the same: activation approximation and decision threshold must be assessed together.

A failure mode arises when polynomial coefficients are fitted to a stale transaction distribution. Fraud patterns drift, and a polynomial that is accurate on historical pre-activations may become poorly calibrated after a shift in merchant mix, customer behavior, or attack strategy. Monitoring pre-activation ranges and score distributions is therefore necessary. Another failure mode is over-regularization of polynomial coefficients, which can smooth the activation so heavily that the model loses sensitivity to nonlinear risk interactions. The regularization parameter $\mu$ in Equation 5 should be selected with attention to both numerical stability and fraud-specific decision loss.

## 7. Technical Discussion

The framework differs from fixed activation replacement in three respects. First, it treats activation degree as a resource to be allocated under a cryptographic depth budget rather than as a uniform hyperparameter. Second, it evaluates approximation quality through a fraud-specific decision objective that includes asymmetric cost and threshold calibration. Third, it provides an error-propagation view that connects local polynomial error to downstream score stability. These elements are important because secure inference systems are often evaluated by aggregate accuracy on standard benchmarks, whereas fraud detection is governed by rare-event sensitivity, alert burden, and cost-sensitive thresholds.

The framework also clarifies the relationship between privacy and model design. Cryptographic privacy protects inputs and weights, but it does not by itself ensure decision quality. A secure protocol that forces a very low-degree activation may produce a model whose decisions differ materially from the plaintext model. Conversely, a high-degree

polynomial may preserve accuracy but exceed the latency budget. The adaptive formulation makes this trade-off explicit. It does not assume that the plaintext network is the only valid target; polynomial-aware training can produce a secure model that is both cryptographically feasible and operationally effective.

Several boundary conditions should be stated. The analysis assumes that the chosen cryptographic protocol provides semihonest security and that implementation-level leakage is addressed separately. It assumes that pre-activation values can be bounded or normalized so that polynomial approximation is meaningful. It assumes that a validation sample is available for threshold calibration and range analysis. These assumptions are standard in secure inference research, but they matter in production fraud systems. The framework is not a substitute for protocol verification, key management, or adversarial testing.

The analysis also has implications for model selection. Shallow networks with fewer nonlinear layers may be easier to evaluate securely, but they may underfit complex fraud patterns. Deep networks may capture richer interactions, but each nonlinear layer consumes depth. The adaptive activation design provides a middle path: depth can be preserved where it matters, while polynomial capacity is concentrated where it most affects fraud decisions. This perspective is more informative than a simple preference for smaller or larger networks.

## 8. Conclusion

Privacy-preserving financial fraud detection requires more than applying a generic secure inference protocol to a generic neural network. The nonlinear activation functions that make neural models expressive are also the operations that make encrypted inference expensive. This article developed an adaptive polynomial activation optimization framework that treats activation degree, Chebyshev coefficients, and decision threshold as coupled design variables. The framework formulates fraud detection as a cost-sensitive decision problem, constrains polynomial activation design by multiplicative depth, and analyzes calibration and error propagation under approximation.

The main conclusion is that secure fraud detection should optimize activation approximation for the fraud decision boundary rather than for average reconstruction error alone. Layer-specific polynomial capacity can be allocated to the parts of the network where approximation error most affects cost-sensitive decisions, while low-degree approximations are retained where they are sufficient. Calibration must accompany activation replacement because polynomial scores can shift the operative threshold even when ranking performance appears stable. Error-propagation analysis provides a conservative but useful guide for identifying layers that require additional capacity or regularization.

The framework offers a technically grounded basis for designing secure neural inference systems in which privacy, fraud cost, and cryptographic feasibility are evaluated jointly. It is applicable to homomorphic encryption, secret-sharing, and hybrid protocols, and it provides a common language for model designers, cryptography engineers, and fraud operations teams. The resulting design perspective is particularly relevant to financial institutions that seek to use neural fraud detection without exposing sensitive transaction data or proprietary model parameters.

## References

1. Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90–113.

2. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10), 4915–4928.

3. Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. arXiv preprint arXiv:1009.6119.

4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). IEEE.

6. Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys, 51(4), 1–35.

7. Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In Proceedings of the 41st Annual ACM Symposium on Theory of Computing (pp. 169–178). ACM.

8. Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014). (Leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory, 6(3), 1–36.

9. Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In Advances in Cryptology – ASIACRYPT 2017 (pp. 409–437). Springer.

10. Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016). CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In Proceedings of the 33rd International Conference on Machine Learning (pp. 201–210). PMLR.

11. Juvekar, C., Vaikuntanathan, V., & Chandrakasan, A. (2018). GAZELLE: A low latency framework for secure neural network inference. In Proceedings of the 27th USENIX Security Symposium (pp. 1651–1669). USENIX Association.

12. Mishra, P., Lehmkuhl, R., Srinivasan, A., Zheng, W., & Popa, R. A. (2020). Delphi: A cryptographic inference service for neural networks. In Proceedings of the 29th USENIX Security Symposium (pp. 2505–2522). USENIX Association.

13. Sanyal, A., Kusner, M. J., Gascón, A., & Kanade, V. (2018). TAPAS: Tricks to accelerate (encrypted) prediction as a service. In Proceedings of the 35th International Conference on Machine Learning (pp. 4494–4503). PMLR.

14. Peng, H., Huang, S., Zhou, T., Luo, Y., Wang, C., Wang, Z., ... & Ding, C. (2023, October). Autorep: Automatic relu replacement for fast private network inference. In 2023 IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 5155-5165). IEEE.

15. Hesamifard, E., Takabi, H., & Ghasemi, M. (2017). CryptoDL: Deep neural networks over encrypted data. arXiv preprint arXiv:1711.05189.

16. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (pp. 807–814). Omnipress.

17. Mohassel, P., & Zhang, Y. (2017). SecureML: A system for scalable privacy-preserving machine learning. In 2017 IEEE Symposium on Security and Privacy (pp. 19–38). IEEE.

18. Bost, R., Popa, R. A., Tu, S., & Goldwasser, S. (2015). Machine learning classification over encrypted data. In Proceedings of the 22nd Network and Distributed System Security Symposium. Internet Society.

19. Chou, E., Beal, J., Levy, D., Yeung, S., Haque, A., & Fei-Fei, L. (2018). Faster CryptoNets: Leveraging sparsity for real-world encrypted inference. arXiv preprint arXiv:1811.09953.

20. Lou, Q., & Jiang, L. (2019). SHE: A fast and accurate deep neural network for encrypted data. In Advances in Neural Information Processing Systems 32. Curran Associates.