Can Deep Learning Tame the Spreadsheet?

Link to the paper—https://arxiv.org/abs/2501.03540

Introduction

A Story of Tables, Transformers, and Triumphs

If you've ever worked in healthcare, finance, retail, or even in a university office, you've dealt with tabular data. It's everywhere—tucked inside spreadsheets, databases, and dashboards. We use it to track everything from loan approvals and patient health records to stock prices and product sales. And while deep learning has taken the world by storm in areas like image recognition and natural language processing, it hits a surprising roadblock in this tabular world. Instead of neural networks, the champions of this domain are older, structured models—especially **Gradient Boosted Decision Trees** (GBDTs). They've been the go-to for years. They're fast. They're accurate. And they're surprisingly tough to beat. But a quiet rebellion

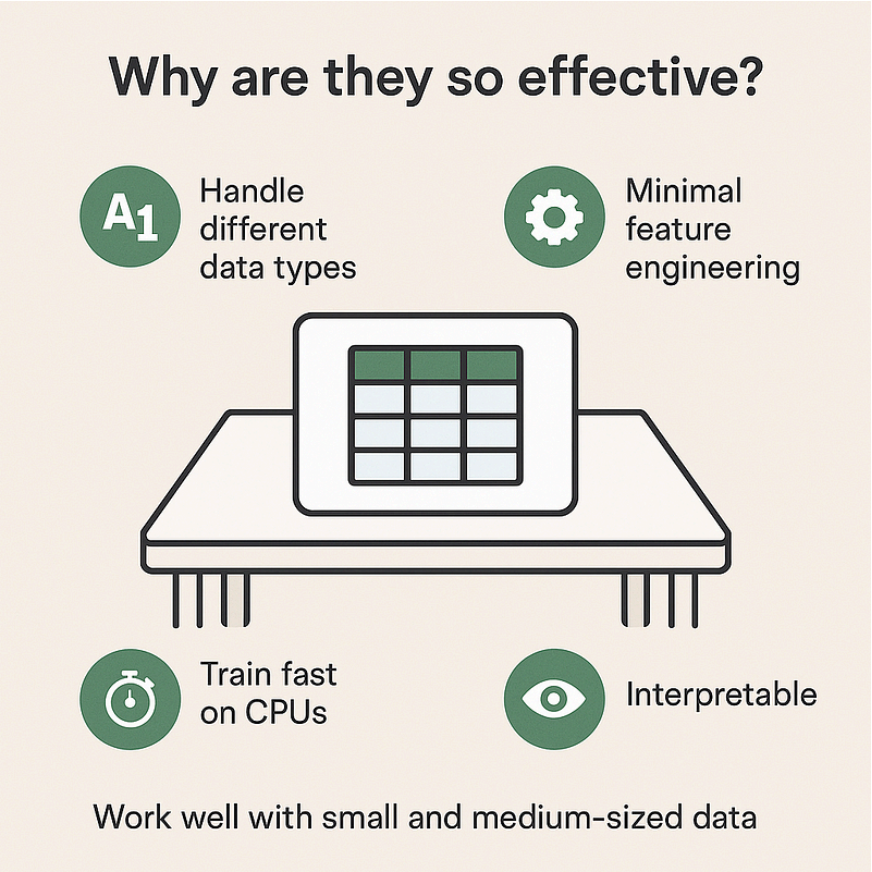
has begun. Recent years have seen deep learning researchers turn their gaze to spreadsheets. New models are being proposed. Transformers are being re-engineered. And foundation models—the giants behind ChatGPT and Stable Diffusion—are trying to prove they can conquer rows and columns too. So the big question is:—Can deep learning finally dethrone decision trees in the world of tabular data?

This article is a story about that battle, that hope, and the architectures that are trying to make it happen.

Why Gradient Boosted Trees Still Dominate Tabular Data?

Tabular data—the kind found in spreadsheets and databases—is everywhere. It powers decisions in finance, healthcare, e-commerce, logistics, and beyond. And when it comes to modeling this kind of data, one class of algorithms still reigns supreme: **Gradient Boosted Decision Trees (GBDTs)**.

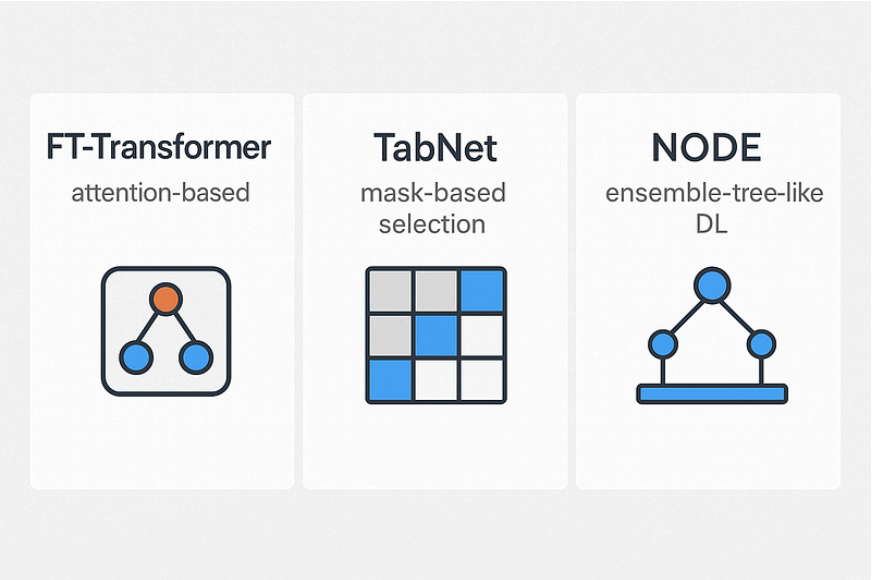
Models like **XGBoost**, **LightGBM**, and **CatBoost** have long been the gold standard for structured data tasks.



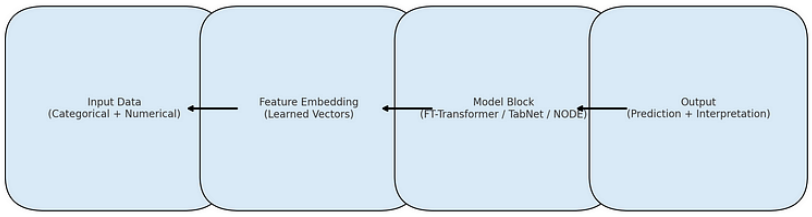
Key Innovations: How Deep Learning is Evolving for Tabular Data

Deep learning didn't give up on tabular data. Instead, it began **adapting**—borrowing tricks from its own past successes in vision and language.

The paper identifies three major architectural shifts that are shaping this next wave of deep models for tabular inputs:



Architecture & Experimental Setup



What Goes Into the Model: Features and Embeddings

Tabular datasets are unique—they mix **categorical and numerical features**, often in messy formats. Unlike pixels or word tokens, table data doesn't have natural order or structure. That's why the **embedding step** is so important in deep models for tabular inputs.

Numerical Features—These are things like age, price, temperature, or income. Most models: Normalize them (e.g., standardize to zero mean and unit variance). Optionally embed them into low-dimensional vectors (similar to projection layers)

Categorical Features—These include names, ZIP codes, job titles, or product categories. These are usually: Converted into **learned**

embeddings, just like how NLP models embed words. The size of the embedding is a hyperparameter—often a function of category count.

Why Embeddings Matter—By turning raw values into dense vectors, models like FT-Transformer and TabNet can: Learn complex interactions between features. Reduce sparsity in large vocabularies. Create a uniform input format for all features.

Inside the Model Block: How Each Architecture Processes the Table

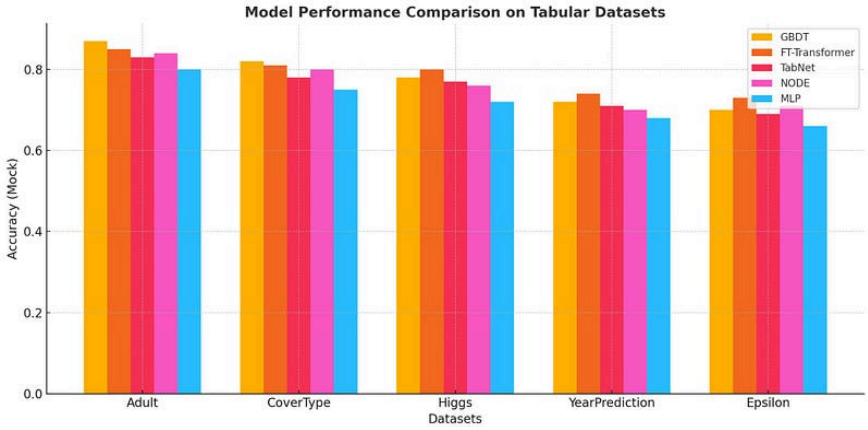
Once the features are embedded into dense vectors, each model architecture applies its own special logic to **learn patterns, relationships, and make predictions**. Here's how they differ:

1. FT-Transformer- *"What if we treat features like tokens in a sentence?"*—Applies **self-attention** over embedded features, similar to how Transformers process words. Learns which features to focus on, and how they interact.
2. TabNet—*"Let's focus only on the features that matter—step by step."*Uses **feature masking** and **sequential attention**. At each decision step, it selects the most relevant features and updates the attention.
3. NODE (Neural Oblivious Decision Ensembles)—"Can we combine the logic of decision trees with the power of neural nets?" Builds an ensemble of differentiable decision trees using neural components. Mimics tree-based splitting in a soft, learnable way.

Benchmarks and Comparisons

Datasets Used:

Dataset	Task Type	Domain
Adult	Binary Classification	Census Income
CoverType	Multiclass Classification	Forest Cover Mapping
Higgs	Binary Classification	Physics Simulation
YearPrediction	Regression	Music Artist Prediction
Epsilon	Binary Classification	Synthetic High-Dim



Comparison Models:

1. Traditional: **XGBoost**, **LightGBM**, **CatBoost**

2. Deep Learning: **FT-Transformer**, **TabNet**, **NODE**, **MLP**

Key Findings:

→ No single winner across all datasets

→ **GBDTs** still dominate small-to-medium datasets.

→ **DL models** perform better when: large-scale data, Categorical features are rich, Embeddings and tuning are optimized.

Conclusion & Future Directions

Foundation models changed how we think about text and vision. This paper shows they're now eyeing a new frontier—the humble table. Whether they succeed depends on how well we reimagine their architectures for structured, messy, real-world data.