# ****Pitch Document: Smart Home IoT Anomaly Detection****

## ****The Problem****

Smart homes are equipped with sensors that generate huge amounts of data every day. While this data has the potential to improve safety, efficiency, and convenience, it is currently underutilized. Most systems still rely on simple threshold-based alerts, which are brittle and often miss complex issues.

Examples of anomalies that matter:

* **Security breach:** a window opens unexpectedly at night.
* **Appliance failure:** a refrigerator’s power consumption suddenly drops to zero.
* **Environmental hazard:** a rapid, unexplained rise in humidity.

# The Solution

I have developed an AI-driven anomaly detection system tailored for smart homes that processes data from multiple sensors (e.g., temperature, humidity, power usage, motion detection, and door states). This system learns the normal daily and weekly patterns of the household. When an anomaly occurs, the system automatically flags it.

The key innovation lies in the **use of deep learning models designed for edge deployment**, making them lightweight and optimized for low-power, low-latency environments. These models—**Convolutional Neural Networks (CNNs)** and **Temporal Convolutional Networks (TCNs)** with **dilated convolutions**—are specifically built to detect anomalies in real-time, without the need for cloud-based computation.

While **LSTM** (Long Short-Term Memory) networks were initially used as a **baseline model**, the use of **CNN** and **TCN** models, particularly with **dilated convolutions**, offers a unique balance of **accuracy** and **deployability**. This makes them ideal for IoT devices, where **low-latency performance, real-time detection, and power efficiency** are critical.

# Demonstrated Technical Expertise

To prove feasibility and showcase my ability to execute, I built the project pipeline entirely from scratch — from data creation to deployable models. Here are the steps I took:

### 1. ****Data Simulation****

* Created a synthetic dataset covering several weeks of smart home activity.
* Modeled **daily and seasonal cycles** to reflect realistic household usage patterns.
* Simulated **sensor drift and dropouts**, ensuring robustness to real-world data issues.
* Incorporated **correlations across signals** (e.g., rising temperature linked with fire alarms).
* Designed **different baselines for different household zones** (kitchen, bedroom, living room).
* Captured **individual lifestyle variations**, recognizing that each household has unique routines.
* Simulated **events of random duration and fluctuating intensity**, making anomalies realistic instead of artificial.

### 2. ****Model Development & Optimization****

**LSTM as Baseline**: Initially used LSTM models as a baseline to model sequential time-series data, which are excellent at capturing long-term dependencies in the data.****

* **CNN and TCN Optimization: Focused on improving performance by incorporating dilated convolutions into CNN and TCN models. Dilated convolutions allow the model to capture long-range dependencies in time-series data without the computational overhead typically associated with LSTMs. This architecture is better suited for edge deployment, offering faster inference times and requiring less memory.**
* **Edge Optimization:**
* **Applied quantization techniques to reduce the model size and increase inference speed, ensuring they can run efficiently on low-power IoT devices.**
* **MLflow for Experiment Tracking: Managed and tracked model experiments with MLflow, enabling easy comparison of model performance and version control.**
* Built a **benchmarking framework** to test and compare model performance, latency and model size.
* Initially, LSTMs outperformed CNNs, but by studying the **observed data patterns**, I **modified CNN architectures** (e.g., dilated convolutions, tuned receptive fields) to **outperform LSTMs**.
* Applied **quantization techniques** to compress models, reducing size and improving inference speed without significant loss in accuracy — critical for edge deployment on smart home devices.
* Implemented **data drift monitoring**, so the system can adapt as household behavior evolves over time.

### 3. ****Experiment Tracking & Version Control****

* Used **MLflow** to track experiments, compare models, and maintain clear model versioning.
* Employed **Git for code version control**, ensuring reproducibility, collaboration readiness, and professional project management.
* Established a robust workflow that ties **code versions, model versions, and results together** — making the system fully auditable and reproducible.

### 3. ****System Engineering****

* Packaged the best-performing models using **Docker and ONNX**, making them portable across environments.
* Created a modular pipeline that allows seamless upgrades of models without disrupting the system.
* Outlined a **monitoring and retraining loop** to ensure the models remain accurate as more data is collected.
* Designed for scalability: the system can move from a single smart home to **thousands of homes** with minimal overhead.

# Business Potential

This project has wide applications across multiple markets:  
- Smart Homes: Improved safety, security, and energy efficiency.  
- Industrial IoT: Predictive maintenance and reduced downtime.  
- Cybersecurity: Detecting unusual patterns in device/network usage.

By demonstrating technical feasibility, I’ve shown that this solution is not only possible but also scalable. With the right support, we can turn this proof-of-concept into a market-ready product.

## 🔧 Technical Readiness Checklist

### ✅ Data & Problem Definition

- [ ] Clearly define the \*\*goal\*\*: “Detect and classify smart home anomalies (power surges, fire, fridge malfunction, etc.) in real time.”

- [ ] Curated \*\*baseline dataset\*\* (simulated + some real IoT data).

- [ ] Show how you can \*\*version & monitor datasets\*\* (DVC, MLflow, Git).

### ✅ Model Development

- [ ] Implemented \*\*baselines\*\*: - LSTM (time series baseline).

- CNN/TCN (efficient sequence models).

- Transformer (for long dependencies).

- [ ] Showed \*\*performance metrics beyond accuracy\*\* (classwise precision/recall, F1, PR curves, confusion matrices).

- [ ] Tackled \*\*class imbalance\*\* with sampling, loss functions, and weighting.

### ✅ Deployment Readiness - [ ] Have a \*\*training-to-inference pipeline\*\* scripted (train → save model → evaluate).

- [ ] Quantization (dynamic/static) for lightweight edge deployment.

- [ ] Export to \*\*ONNX\*\* for device compatibility. - [ ] Basic \*\*data drift monitoring\*\* implemented.

### ✅ Experiment Tracking

- [ ] Logging with \*\*MLflow\*\* (metrics, artifacts, confusion matrix, PR curves).

- [ ] W&B or similar tool for run visualization.

- [ ] Comparison of model sizes (PyTorch, quantized, ONNX). ### ✅ Prototype Demo

- [ ] Notebook / script to show \*\*live evaluation\*\*: load model, run predictions, visualize drift/confusion matrix.

- [ ] Possibly simulate a \*\*streaming IoT feed\*\* (from CSV or MQTT mock). ---

## 📈 Business & Pitch Readiness Checklist

### ✅ Problem & Opportunity

- [ ] Pain point: IoT devices generate lots of data, failures (fire alarm, fridge) go unnoticed.

- [ ] Market: Smart home + insurance + energy management industries.

- [ ] Value: Reduced risk, automated alerts, better energy efficiency.

### ✅ Competitive Edge

- [ ] State-of-the-art anomaly detection (LSTM/TCN/Transformer mix).

- [ ] Lightweight models for \*\*edge deployment\*\* (not just cloud).

- [ ] Data drift monitoring → system improves over time.

### ✅ Traction / Proof of Work

- [ ] Prototype running with quantized models.

- [ ] Clear demonstration that \*\*you can build models without external help\*\*.

- [ ] GitHub repo + demo notebook to show technical maturity.

### ✅ Ask (Funding Justification)

- [ ] Funding needed for: - Access to \*\*real-world smart home datasets\*\* (partnerships, devices).

- \*\*Cloud deployment pipeline\*\* (scaling inference, storage, dashboards).

- Business development & partnerships (insurance, smart home OEMs).

- [ ] Clear \*\*timeline to MVP\*\* with funding (e.g., “With $X, we’ll go from prototype to pilot deployment in 6 months”).

# Roadmap

The journey from prototype to product involves three main stages:  
1. Prototype (✅ Completed): Demonstrated AI-driven anomaly detection with synthetic data.  
2. Pilot: Deploy the system in a small number of smart homes to validate in real environments.  
3. Scale: Expand to larger deployments, integrate with consumer smart home platforms, and add continuous monitoring and retraining.

# Closing

In summary, I have the technical expertise to not only design advanced AI models but also make them deployable in real-world scenarios. This project demonstrates both my execution capability and the market opportunity in smart anomaly detection. I am excited to take this vision forward and welcome your support in making smart homes safer, smarter, and more reliable.