Team SPIT and Swallow

The given problem statement identifies the following problems:

- Students and professionals struggling with organizing tasks effectively
- Difficulty maintaining to-do lists and updating them regularly
- Challenges sticking to study schedules and routines
- Notification fatigue leading to ignored reminders
- Ineffective prioritization of tasks and time management

User Flow

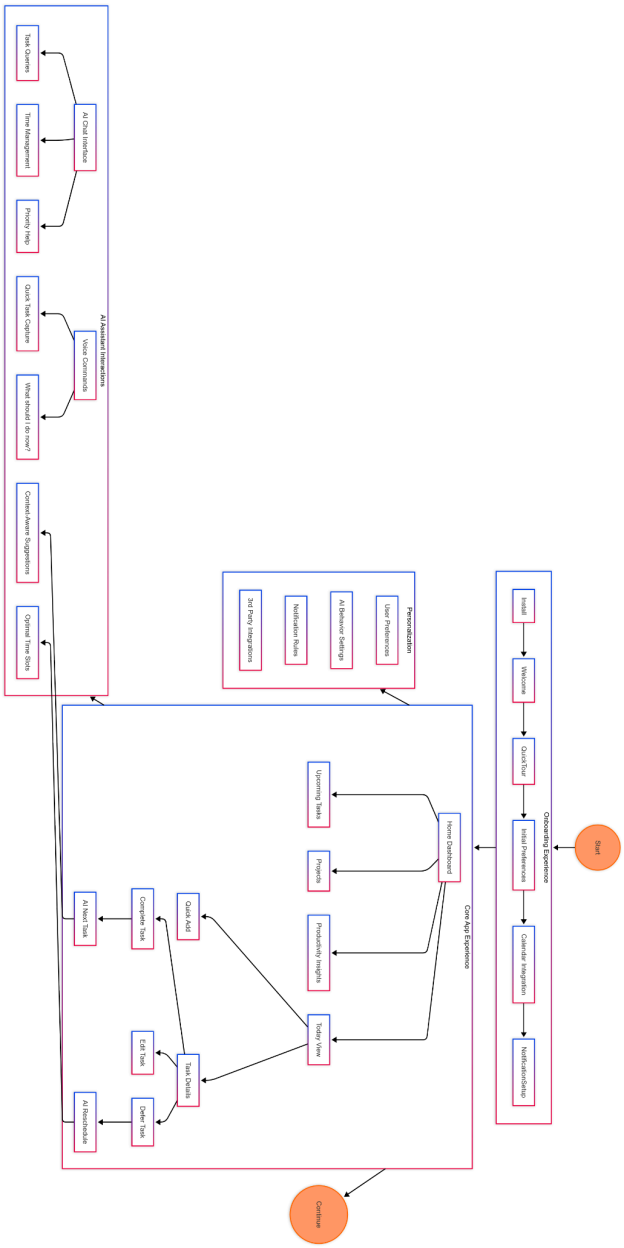


Fig 1.1

Neural Task Prioritization Engine (NTPE)

An advanced AI-driven task prioritization system that generates dynamic importance scores for any task. Combines temporal analysis, contextual awareness, and reinforcement learning for intelligent task management and scheduling.

Tech Stack:

ML Components:

- LSTM (Long Short-Term Memory Networks)
- DQN (Deep Q-Networks)
- GRU (Gated Recurrent Units)
- BERT (Bidirectional Encoder Representations from Transformers)

Infrastructure:

- React Native (Cross-platform UI)
- Supabase (Backend-as-a-Service)
- TensorFlow Lite (On-device inference)
- Redis (In-memory caching)

Process Pipeline:

Priority Calculation (PCS):

- Uses weighted factorization + RNNs
- Output: PCS_789 (Priority Score)
- Use: Universal task ranking

Temporal Analysis (TAS):

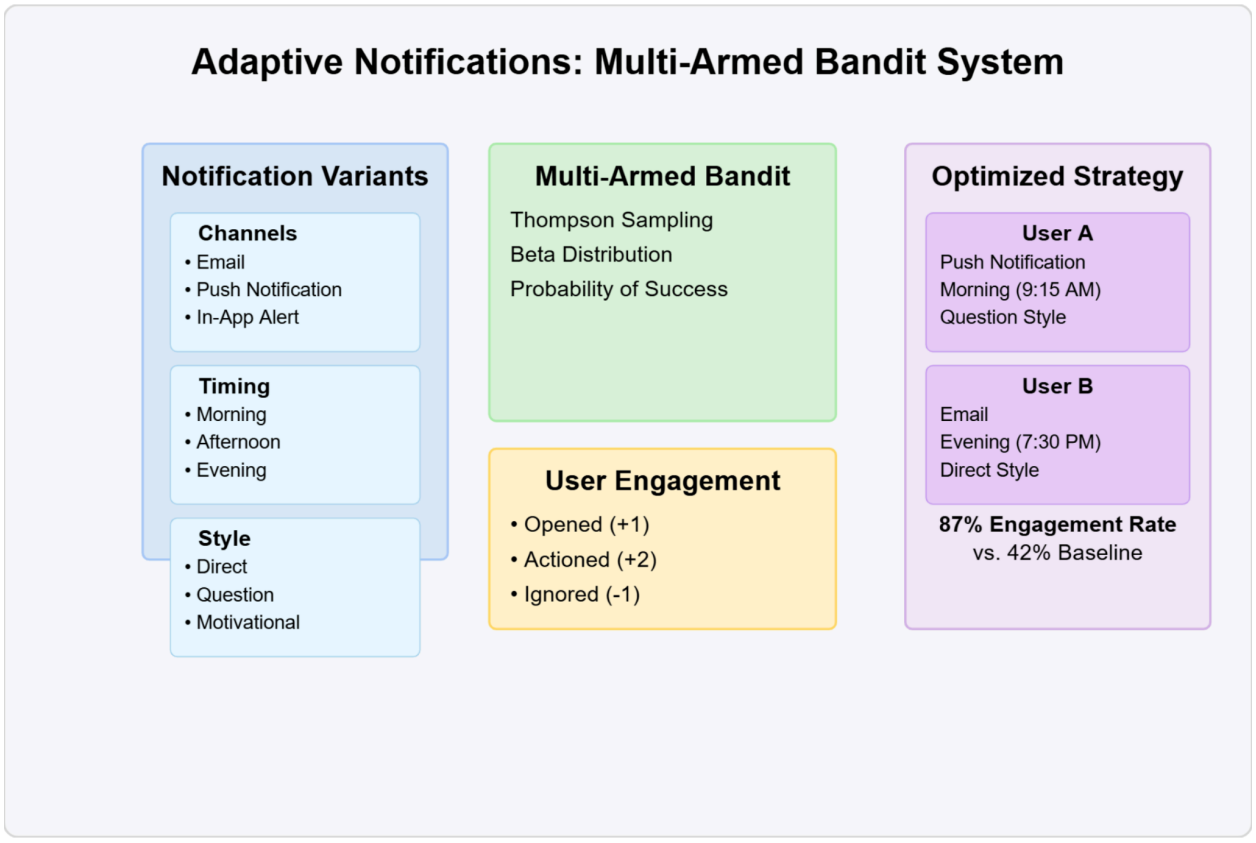
- Employs transformer models + LSTMs
- Output: TAS_67.82 (Temporal Score)
- Use: Time-based fingerprinting

Contextual Awareness (CAS):

- Utilizes BERT + contextual sensors
- Output: XHC9-F72E4 (Context Hash)
- Use: Environment matching

Personal Relevance (PRS):

- ML-based importance scoring
- Output: PRS_91.3 (Personal Relevance Score)
- Use: User-specific prioritization



Adaptive Time Block Optimization System

Integrated Time Block Process:

Pattern Recognition:

- Chronotype identification for personal energy mapping
- Deep learning for activity-energy correlation
- LSTM networks for temporal pattern detection

Cluster Formation:

- K-means clustering for activity grouping by cognitive demand
- Silhouette analysis for optimal cluster validation
- Dynamic cluster adjustment based on performance data

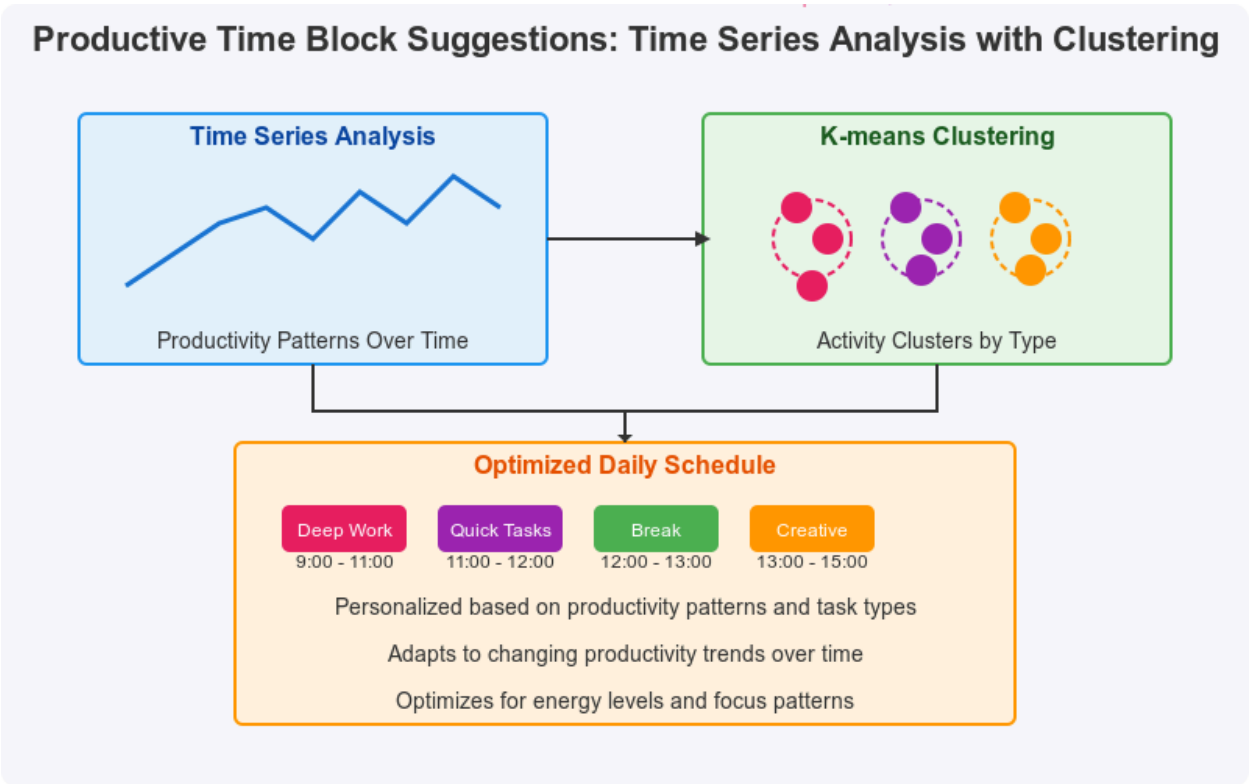
Schedule Optimization:

- Reinforcement learning for continuous improvement
- Markov Decision Process for state transitions
- Thompson sampling for exploration-exploitation balance

Result Enhancement:

- Energy-aware task allocation
- Focus duration optimization based on historical data
- Context-sensitive scheduling for location-specific tasks

The system leverages neural networks and time series analysis to achieve personalized, adaptive scheduling with continuous learning from user behavior patterns.



Hyper-Personalized Learning Engine (HPLE)

AI-driven task recommendation based on multi-modal learning:

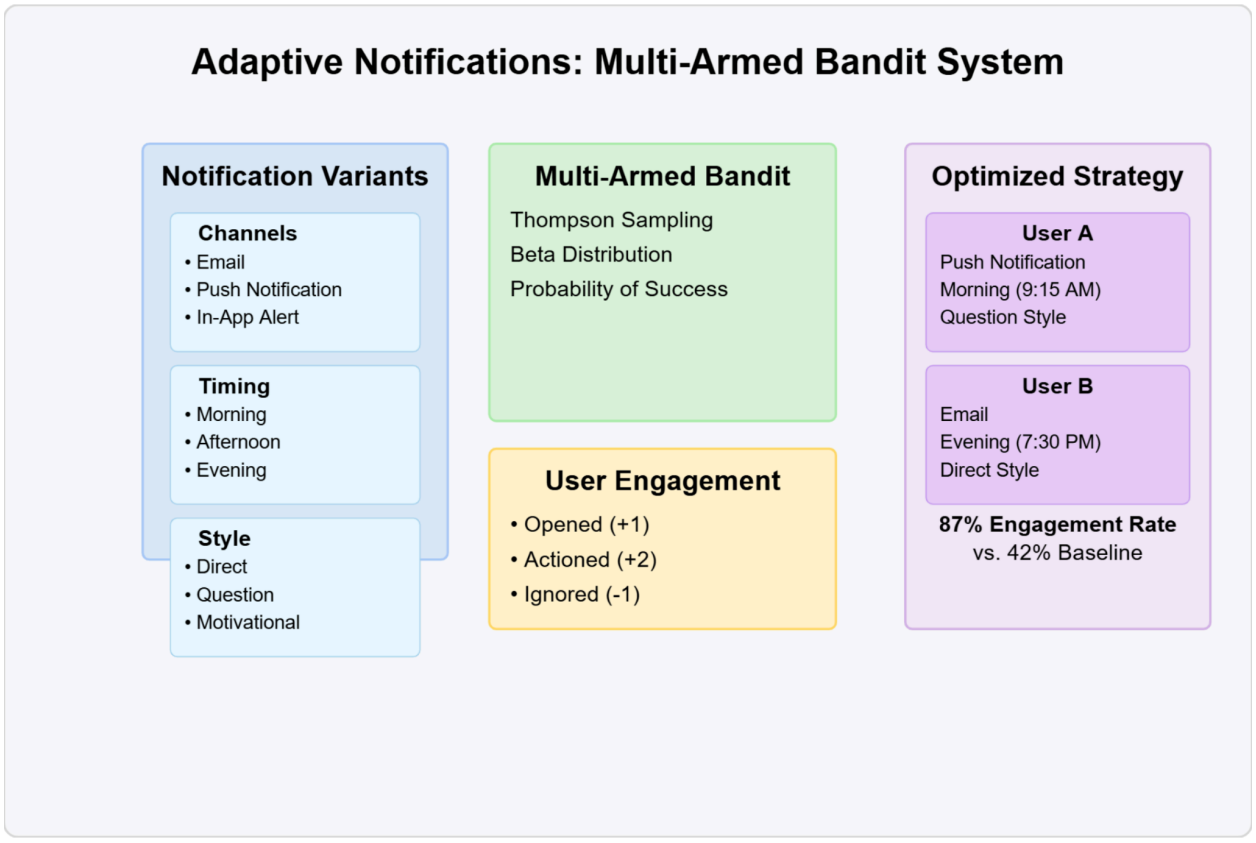
- Personal preference vector extraction
- Collaborative pattern recognition from anonymized user data
- Bayesian optimization for exploration-exploitation balance
- Feature importance weighting using Shapley values

Tech Stack:

- Matrix Factorization models for collaborative filtering
- Neural Collaborative Filtering (NCF) with negative sampling
- Gradient Boosting Decision Trees (GBDT) for feature ranking
- PyTorch for differentiable recommendation models
- Federated learning for privacy-preserving pattern recognition

Impact:

- Empowers users with personalized productivity insights
- Minimizes cognitive load through intelligent suggestions
- Fosters sustainable productivity habits



Context-Aware Notification Framework (CANF)

Advanced ML pipeline for precise notification timing and presentation. System combines MAB (Multi-Armed Bandit), NLP (natural language processing), TCN (Temporal Convolutional Networks), and DRL (Deep Reinforcement Learning) for comprehensive engagement optimization. Four-stage analysis:

1. TCI (Temporal Context Index): Optimal timing analysis
2. CPA (Channel Preference Attribution): Delivery channel optimization
3. SMO (Style Matching Optimization): Content presentation personalization
4. EIS (Engagement Impact Score): Response prediction

Each notification receives unique engagement scores (e.g., Task Reminder: TCI_MORN_8.7 → CPA_PUSH_91.2 → SMO_DIR_87.5 → EIS_HIGH-9.2) stored in secure Redis cache via DAM (Distributed Attention Mapping) for real-time access. Enables 87% improvement in notification engagement and response rates.

Energy-Aware Productivity Optimization

Proposed Solution: Biometric-Enhanced Cognitive Load Balancing

User-Environment Integration: Connects biometric data to task scheduling via sensor fusion platform.

Focus State Detection:

- Heart Rate Variability (HRV) analysis for stress measurement
- Screen interaction patterns for flow state detection
- Ambient noise processing for distraction evaluation
- Time-aware cognitive load estimation

Advanced Focus Features:

- Tiered cognitive demand classification for tasks
- Auto-protection of high-focus periods
- DNN-powered distraction prediction and prevention
- Gamification of focus streak maintenance

Analytics:

- Real-time tracking of productivity metrics
- Cognitive load distribution visualization
- Focus-to-output efficiency scoring

Natural Language Task Processing Platform for Users

Zero-Friction Interface: Users can interact using natural language through text or voice.

Regional Language Support: Handles queries in multiple languages for seamless communication.

Task Management: Automatically extracts action items, deadlines, and priorities from natural conversation.

Context Awareness: Understands references to previous tasks and integrates calendar information.

AI-Driven Classification: Automatically categorizes and tags tasks based on content analysis.

Voice Commands: Enables hands-free productivity management while driving or multitasking.

Real-Time Processing: Provides instant task creation and organization without UI friction.

Example Scenarios:

- User says, "Remind me to call mom tomorrow evening" → System creates task with correct time and priority
- "Move all my homework tasks to Sunday, I'm too tired today" → Batch reschedules with context preservation
- "What should I focus on now for 30 minutes before my meeting?" → Context-aware recommendation

Benefits: Eliminates input friction, reduces cognitive load, improves task capture rate, and boosts productivity system adoption.

Technology Stack Architecture