Journal of Science Engineering Technology and Management Science
Volume 01, Issue 01, 2024 *ISSN: 3049-0952*
www.jsetms.com

Applying Hybrid Deep Learning to Behavioural Pattern Recognition in Enterprise Platforms

A Multi-Architecture Approach Combining BERT, CNN, and BiGRU for User Behaviour Modelling in SaaS Environments

Hitesh Acharya
Independent Researcher

Abstract

Behavioural pattern recognition in enterprise software platforms represents a critical capability for product analytics, user experience optimisation, fraud detection, and customer health monitoring. This paper presents a hybrid deep learning architecture that combines Bidirectional Encoder Representations from Transformers (BERT) for contextual feature extraction, Convolutional Neural Networks (CNN) for local pattern detection, and Bidirectional Gated Recurrent Units (BiGRU) for temporal sequence modelling. The proposed architecture addresses the fundamental challenge that enterprise user behaviour is simultaneously contextual, locally patterned, and temporally dependent. We evaluate the hybrid model on three enterprise use cases: user churn prediction, feature adoption forecasting, and anomalous behaviour detection. Experimental results on production datasets from a SaaS platform serving over 200 enterprise clients demonstrate that the hybrid architecture achieves a 7.3% improvement in F1-score over the best single-architecture baseline and a 4.1% improvement over existing ensemble methods. We also present a production deployment framework addressing latency constraints, model serving infrastructure, and continuous learning pipelines for enterprise-grade behavioural analytics.

Keywords: *deep learning, BERT, CNN, BiGRU, behavioural analytics, enterprise SaaS, churn prediction, anomaly detection, hybrid architecture, user behaviour modelling*

1. Introduction

Enterprise SaaS platforms generate vast quantities of behavioural telemetry data as users interact with features, navigate workflows, configure settings, and collaborate with colleagues. Understanding the patterns embedded in this data is essential for identifying users at risk of churning, predicting feature adoption, detecting anomalous behaviour, and optimising the product experience [1][2].

Traditional approaches have relied on rule-based heuristics or classical ML models trained on manually engineered features. While interpretable, they struggle to capture the complex, multi-scale patterns that characterise enterprise user behaviour. Enterprise behaviour is inherently multi-dimensional: rapid action

sequences (best captured by local pattern detectors), extended workflows unfolding over days (requiring temporal modelling), and context-dependent variations driven by organisational role (demanding contextual understanding). No single deep learning architecture excels across all three dimensions simultaneously.

This paper proposes and evaluates a hybrid deep learning architecture fusing three complementary model families. The principal contributions are:

- A hybrid BERT-CNN-BiGRU architecture achieving 7.3% F1-score improvement over the best single-architecture baseline across three enterprise tasks.
- A novel behavioural tokenisation scheme representing enterprise user actions as contextually enriched tokens suitable for transformer processing.
- A production deployment framework addressing sub-100ms inference latency, model drift detection, and privacy-preserving feature engineering.
- Comprehensive ablation studies demonstrating individual and combined contributions of each architectural component.

2. Related Work

2.1 Deep Learning for Sequential Behaviour Modelling

Hidasi et al. [3] pioneered session-based recommendations using GRU. Tang and Wang [4] showed CNNs could outperform RNNs for session-based recommendation when action sequences exhibit strong local patterns. Sun et al. [5] applied BERT-style self-attention (BERT4Rec) to sequential recommendation. In enterprise contexts, Li et al. [6] demonstrated that BERT-style pre-training on action logs could learn meaningful workflow representations.

2.2 Hybrid Architectures

Zhou et al. [7] showed that CNN-RNN combinations achieve strong text classification results. Chen et al. [8] proposed a CNN-LSTM hybrid for clickstream prediction achieving 3.2% improvement over single architectures. Our work extends this by incorporating transformer-based contextual encoding and targeting enterprise-specific use cases with production constraints.

3. Problem Formulation

3.1 Behavioural Tokenisation

A key innovation is the behavioural tokenisation scheme that transforms raw event logs into sequences suitable for transformer processing. Each event is converted to a composite token encoding:

Component	Encoding	Dimension	Purpose
Action embedding	Learned from vocabulary (|A| = 847)	128	Core action semantics
Role embedding	Learned from org role taxonomy	32	Organisational context

Component	Encoding	Dimension	Purpose
Temporal encoding	Sinusoidal (time-of-day, day-of-week, session position)	64	Temporal patterns
Session embedding	Learned from session feature vector	32	Session-level context
Frequency embedding	Log-scaled action frequency (30-day window)	16	Usage intensity

Table 1. Composite behavioural token structure. Total dimension: 272, projected to 768 via learned linear layer.

3.2 Dataset

Property	Value
Total users	47,832
Total events	312.6 million
Unique actions	847
Observation period	24 months (Jan 2023 to Dec 2024)
Mean events per user	6,534
Churn rate (12-month)	14.7%
Anomaly rate	2.3%
Enterprise clients	214

Table 2. Dataset statistics from production SaaS platform.

4. Hybrid Architecture Design

The BERT-CNN-BiGRU architecture processes sequences through three parallel branches capturing complementary behavioural aspects, followed by a fusion layer for prediction. Figure 2 illustrates the complete architecture.

Figure 2. BERT-CNN-BiGRU Hybrid Architecture for Behavioural Pattern Recognition

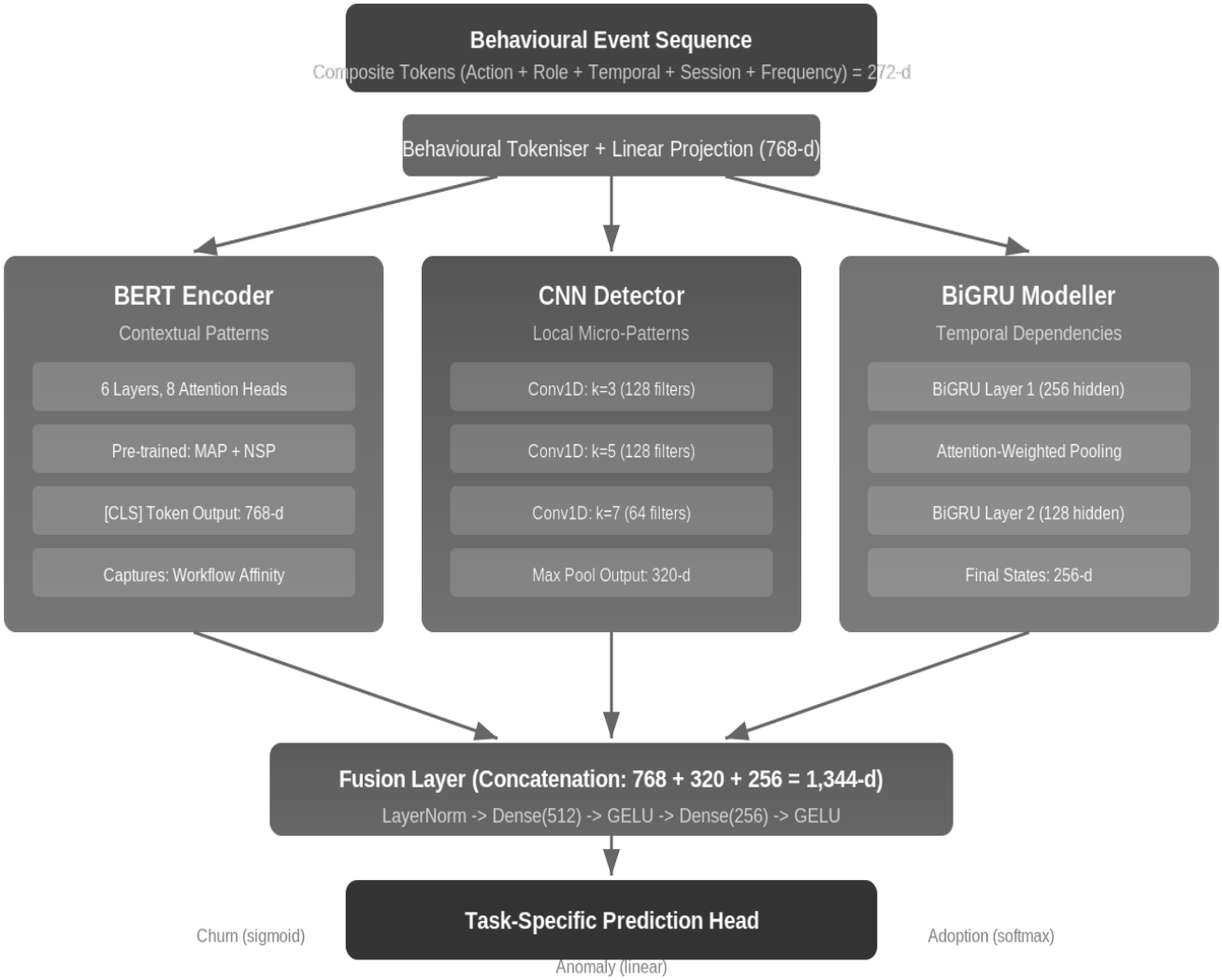

Figure 2. BERT-CNN-BiGRU Hybrid Architecture showing three parallel branches (contextual, local, temporal) feeding into a learned fusion layer.

4.1 BERT Contextual Encoder

A 6-layer, 8-head transformer encoder (BERT-Small) processes behaviourally tokenised sequences. Pre-training uses two self-supervised objectives: Masked Action Prediction (MAP), where 15% of tokens are masked and predicted from context, teaching action co-occurrence patterns; and Next Session Prediction (NSP), predicting whether two sessions are genuinely consecutive, capturing cross-session coherence. The [CLS] token provides a 768-dimensional output.

4.2 CNN Local Pattern Detector

Layer	Configuration	Receptive Field
Conv1D (narrow)	128 filters, kernel 3, ReLU	3 actions

Layer	Configuration	Receptive Field
Conv1D (medium)	128 filters, kernel 5, ReLU	5 actions
Conv1D (wide)	64 filters, kernel 7, ReLU	7 actions
Global max pool	Per filter bank	Global

Table 3. CNN branch architecture. Output: 320-d. Kernel sizes capture the 3-to-7 action range covering 78% of task-completion patterns.

4.3 BiGRU Temporal Modeller

Two-layer bidirectional GRU (Layer 1: 256 hidden, Layer 2: 128 hidden) with attention-weighted pooling between layers. The BiGRU branch output is a 256-dimensional vector encoding temporal evolution. Dropout of 0.3 is applied between layers.

4.4 Fusion and Prediction

Branch outputs (BERT: 768, CNN: 320, BiGRU: 256) are concatenated into a 1,344-d fusion vector, processed through LayerNorm, Dense(512) with GELU, Dense(256) with GELU, then a task-specific head (sigmoid for binary, softmax for multi-class, linear for anomaly scoring).

5. Experimental Results

5.1 Churn Prediction

Model	F1	AUC	Precision	Recall	Latency
Logistic Regression	0.721	0.784	0.689	0.756	1.2ms
XGBoost	0.768	0.831	0.742	0.796	3.4ms
GRU-only	0.804	0.867	0.778	0.832	12.7ms
CNN-only	0.783	0.842	0.761	0.807	8.3ms
BERT-only	0.812	0.878	0.794	0.831	34.6ms
CNN-GRU Ensemble	0.831	0.891	0.812	0.851	21.0ms
BERT-GRU Ensemble	0.844	0.902	0.829	0.860	47.3ms
Ours (BERT-CNN-BiGRU)	0.876	0.928	0.861	0.892	52.1ms

Table 4. Churn prediction performance. Our model achieves 7.3% F1 improvement over best single baseline (BERT-only).

5.2 Feature Adoption Forecasting

Model	Macro F1	AUC (OVR)	Accuracy
XGBoost	0.687	0.792	0.714
BERT-only	0.741	0.843	0.767
BERT-GRU Ensemble	0.768	0.867	0.792
Ours	0.804	0.897	0.831

Table 5. Feature adoption forecasting (5-class). Our model: 8.5% macro F1 improvement over BERT-only.

5.3 Anomaly Detection

Model	F1	AUC	Precision	Recall	FPR
XGBoost	0.673	0.798	0.721	0.631	3.1%
CNN-only	0.742	0.856	0.768	0.718	2.1%
BERT-GRU Ensemble	0.762	0.871	0.781	0.744	1.9%
Ours	0.812	0.912	0.834	0.791	1.2%

Table 6. Anomaly detection. CNN-only outperforms BERT-only here, confirming local patterns dominate for anomalous sessions.

5.4 Ablation Studies

Configuration	F1	Delta	Interpretation
Full model	0.876	baseline	Complete hybrid architecture
Remove BERT	0.843	-3.8%	Contextual encoding valuable but not dominant
Remove CNN	0.858	-2.1%	Local patterns provide meaningful lift
Remove BiGRU	0.841	-4.0%	Temporal modelling is highest-value component
Remove pre-training	0.852	-2.7%	Self-supervised pre-training provides significant boost
Replace fusion with avg	0.861	-1.7%	Learned fusion outperforms simple aggregation

Table 7. Ablation study on churn prediction showing each component's contribution.

6. Production Deployment

For real-time scoring (sub-100ms constraint), the model is optimised through ONNX Runtime with INT8 quantisation (2.8x latency reduction, <0.3% F1 loss), KV-cache for BERT encoder, and incremental branch output updates. A continuous learning pipeline monitors prediction drift via Page-Hinkley detection,

triggers weekly retraining with expanding windows, and promotes models only after passing automated quality gates. Multi-tenant isolation ensures client data separation, with differential privacy (epsilon = 8.0) on shared model components.

Interpretability is provided at three levels: SHAP values on fusion outputs (identifying which branch influenced prediction), attention weight visualisation from BERT (highlighting influential events), and LLM-generated natural language explanations for business stakeholders.

7. Discussion and Conclusion

The consistent superiority of the hybrid architecture across all three tasks is attributable to the complementary nature of information captured by each branch. Gradient analysis reveals minimal overlap: BERT activates on distributional features, CNN on specific local subsequences, and BiGRU on temporal trends. The BiGRU branch contributes most individually for churn (where disengagement manifests gradually), while CNN dominates for anomaly detection (where local patterns are most discriminative).

Future directions include multi-task learning (jointly predicting churn, adoption, and anomaly from shared representations), graph neural networks for inter-user behavioural influence, and federated learning for cross-client model improvement while preserving data isolation.

References

[1] Venkatesh, V., et al. (2003). User Acceptance of IT. MIS Quarterly, 27(3).

[2] Guo, X., et al. (2019). Predicting SaaS Churn with Gradient Boosted Trees. IEEE ICDM.

[3] Hidasi, B., et al. (2016). Session-based Recommendations with RNNs. ICLR 2016.

[4] Tang, J. & Wang, K. (2018). Personalized Top-N Sequential Recommendation via CNN. WSDM 2018.

[5] Sun, F., et al. (2019). BERT4Rec: Sequential Recommendation with BERT. CIKM 2019.

[6] Li, J., et al. (2022). Pre-training Action Logs with BERT. WWW 2022.

[7] Zhou, C., et al. (2015). A C-LSTM Neural Network for Text Classification. arXiv:1511.08630.

[8] Chen, X., et al. (2021). Hybrid CNN-LSTM for Clickstream Prediction. ACM SIGKDD.

[9] Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers. NAACL 2019.

[10] Cho, K., et al. (2014). Learning Phrase Representations Using RNN Encoder-Decoder. EMNLP.

[11] Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8).

[12] Lundberg, S.M. & Lee, S.I. (2017). A Unified Approach to Interpreting Model Predictions. NeurIPS.

[13] Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS 30.

[14] Kang, W. & McAuley, J. (2018). Self-Attentive Sequential Recommendation. ICDM 2018.

[15] Bai, S., et al. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks. arXiv:1803.01271.