

**LAYOUT-AGNOSTIC CHANGE POINT DETECTION FOR ENHANCED HUMAN ACTIVITY
RECOGNITION IN SMART HOMES**

An Undergraduate Thesis
Presented to
The Academic Faculty

By

Tyler Stennett

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science in Computer Science in the
Georgia Institute of Technology
College of Computing

Georgia Institute of Technology

May 2025

**LAYOUT-AGNOSTIC CHANGE POINT DETECTION FOR ENHANCED HUMAN ACTIVITY
RECOGNITION IN SMART HOMES**

Thesis committee:

Dr. Thomas Plötz
School of Interactive Computing
Georgia Institute of Technology

Dr. Sonia Chernova
School of Interactive Computing
Georgia Institute of Technology

Date approved: April 22, 2025

ACKNOWLEDGMENTS

I am deeply grateful to my family for their unwavering support throughout my academic journey. To my late father, whose memory continues to motivate and inspire me, and to my mother and sister, thank you for your love, encouragement, and belief in me.

I would like to extend my sincere thanks to my faculty mentor, Professor Thomas Plötz, for his invaluable guidance, expertise, and patience. I am also thankful to my second reader, Professor Sonia Chernova, for her kindness and thoughtful feedback.

Finally, I would like to express my heartfelt appreciation to Sourish Dhekane and Megha Thukral for their generous support and insightful feedback throughout the project. Your contributions made a meaningful difference.

SUMMARY

Human Activity Recognition (HAR) in smart home environments critically depends on accurate temporal segmentation. However, most existing pipelines rely on fixed, overlapping windows that often obscure activity boundaries, degrading downstream classifier performance. This thesis introduces two layout-agnostic change point detection (CPD) frameworks, **TAD** and **TCPC**, which achieve state-of-the-art performance across diverse smart home domains.

TAD (Textual Descriptions of Sensor Triggers-based Activity Distribution) Change Point Detection uses semantically rich sensor descriptions (via TDOST) and a BiLSTM HAR classifier. By adapting the Snip-Snap paradigm, it detects change points through divergences in predicted activity distributions between adjacent, non-overlapping windows.

TCPC (Textual Descriptions of Sensor Triggers-based Contrastive Predictive Coding) Change Point Detection eliminates the requirement for labeled data. A contrastive predictive coding (CPC) model learns domain-invariant embeddings by predicting future latent states from past context, with change points identified through cosine similarity comparisons across consecutive windows.

Both methods were evaluated on four CASAS smart homes (Aruba, Milan, Kyoto, and Cairo), and were shown to consistently outperform statistical (SEP) and embedding-based (CCS) baselines, with gains ranging from 1.20% to 16.45% in the geometric mean of the TPR and TNR. When integrated with a BiLSTM classifier, these segmenters also yielded downstream classification improvements of 1.66% to 8.18% in Macro F1 score over the prominent sliding window approach.

By using lightweight sentence-transformer embeddings and modest recurrent architectures, both frameworks are well-suited for real-time deployment. This thesis further outlines directions for advancing HAR through transformer-based segmenters, richer datasets, and hardware-efficient models. Overall, the work demonstrates that intelligent, representation-driven segmentation is both necessary and practical for reliable activity recognition in smart homes.

LIST OF TABLES

3.1	Summary of the Materials Used in the Project.	9
4.1	Average Geometric Mean for Each Method Across the Target Datasets	18
A.1	TAD Change Point Detection Accuracy Across Similarity Metrics and Context Window Sizes (Exact vs. ± 5 s Tolerant) Evaluated on the Aruba Dataset.	25
A.2	Hyperparameter Tuning Results for TCPC: Exact Change Point Detection Performance on the Aruba Dataset.	26
A.3	Hyperparameter Tuning Results for TCPC: ± 5 s Tolerance Change Point Detection Performance on the Aruba Dataset.	27

LIST OF FIGURES

3.1	The supervised approach to data segmentation	10
3.2	The unsupervised approach to data segmentation	13
4.1	Exact Change Point Detection TPR and FPR	16
4.2	Exact Change Point Detection Geometric Mean	17
4.3	± 5 s Tolerance Change Point Detection TPR and FPR	17
4.4	± 5 s Tolerance Change Point Detection Geometric Mean	18
4.5	Classification Macro F1 Score	19

LIST OF ACRONYMS

AI	Artificial Intelligence
AUC	Area Under Curve
BiLSTM	Bidirectional Long Short-Term Memory
BT	Barlow Twins
BYOL	Bootstrap Your Own Latent
CAC	Corrected Area Curve
CASAS	Cognitive Assistance for Smart Apartments
CCS	Context Cosine Similarity Score
CNN	Convolutional Neural Network
CPC	Contrastive Predictive Coding
CPD	Change Point Detection
E2USD	Efficient and Unsupervised State Detection
FLOSS	Fast Low-Cost Online Semantic Segmentation
FPR	False Positive Rate
GRU	Gated Recurrent Unit
HAR	Human Activity Recognition
HMM	Hidden Markov Model
InfoNCE	Information Noise-Contrastive Estimation
LSTM	Long Short-Term Memory
NB	Naive Bayes
PLA	Piece-wise Linear Approximation
PLR	Piece-wise Linear Representation
ROC	Receiver Operating Characteristic
RuLSIF	Relative Unconstrained Least-Squares Importance Fitting
SEP	Separation
SHiB	Smart Home in a Box
SimCLR	Simple Framework for Contrastive Learning

SupCon Supervised Contrastive

SVM Support Vector Machine

SWAB Sliding Window and Bottom-Up

TAD Textual Descriptions of Sensor Triggers-based Activity Distribution Change Point Detection

TCPC Textual Descriptions of Sensor Triggers-based Contrastive Predictive Coding Change Point Detection

TDOST Textual Descriptions of Sensor Triggers

TNR True Negative Rate

TPR True Positive Rate

TS-CP² Time Series Change Point Detection Based On Contrastive Predictive Coding

TABLE OF CONTENTS

Acknowledgments	iii
Summary	iv
List of Tables	iv
List of Figures	vi
List of Acronyms	vii
Chapter 1: Introduction	1
Chapter 2: Literature Review	3
2.1 Human Activity Recognition	3
2.2 Data Segmentation Techniques	4
2.3 Contrastive Learning	6
Chapter 3: Frameworks & Methods	9
3.1 Materials	9
3.2 Methodology	9
3.2.1 Approach 1: Supervised Learning (TAD)	10
3.2.2 Approach 2: Self-supervised Learning (TCPC)	12
Chapter 4: Results	15
4.1 Change Point Detection and Transferability	15
4.1.1 Exact Change Point Detection	16
4.1.2 Tolerant Change Point Detection	17

4.2	Downstream Classification	18
Chapter 5: Discussion	20
5.1	Limitations	20
5.2	Future Work	20
5.3	Significance	21
Chapter 6: Conclusion	22
Appendices	23
Appendix A:	Change-Point Detection: Metric Selection and Hyperparameter Optimization	24
References	28

CHAPTER 1

INTRODUCTION

The continuous evolution of integrated devices, coupled with the rapid rise of Artificial Intelligence (AI), has led to a significant increase in the demand for smart home products. In fact, the global smart home market is projected to grow from \$101.01 billion in 2023 to a substantial \$633.20 billion by 2032 [1]. This surge in popularity has ignited interest in Human Activity Recognition (HAR), where embedded sensors in smart home environments are used to monitor and classify human activities [2]. Research in the field of HAR has addressed several critical challenges, including the representation of sensor data, the development of effective model architectures for accurate activity recognition, and generalizability across diverse target domains. However, segmentation methods remain comparatively underdeveloped.

Most HAR systems rely on fixed-length, overlapping windows to segment sensor data before feeding it into activity identification models [3, 4, 5, 6]. This naive windowing approach introduces two major drawbacks. First, overlapping windows may contain data from different activities, thereby mixing contextual signals and diluting predictive accuracy. Second, fixed window sizes may truncate longer activities, leading to incomplete representations and reduced performance. Although several statistics-based and embedding-based segmentation methods have been proposed to address these issues [7, 6, 8, 9], they often fail to generalize effectively across diverse smart home settings or accurately decipher complex human activity readings. To overcome these limitations, my research explores advanced techniques from temporal data segmentation and representation learning to improve the detection of change points, or moments in time series data where underlying patterns shift significantly [10]. By identifying these change points more accurately, the goal is to enhance segmentation quality and, consequently, improve the overall performance of HAR systems.

An early approach to Change Point Detection (CPD), known as the Snip-Snap approach, was developed by Plotz *et al.* [11]. This method used non-overlapping sequential windows called “Snip” and “Snap,” applying predictive models on the “Snip” window and then comparing the similarity of its output with the “Snap” window to detect change points. Building on this concept, I propose two novel methods tailored to the challenges of HAR: **Textual Descriptions of Sensor Triggers-based Activity Distribution (TAD) Change Point Detection**, and **Textual Descriptions of Sensor Triggers-based Contrastive Predictive Coding (TCPC) Change Point Detection**. In TAD, sensor data is embedded using Textual Descriptions of Sensor Triggers (TDOST) and sentence transformers, which has demonstrated improvements in transfer learning by standardizing sensor data structure with contextual information textually before creating embeddings [4]. A

pretrained Long Short-Term Memory (LSTM) recurrent neural network will soft-classify embeddings from two non-overlapping windows, and the dot product will measure their difference, with a low output indicating highly dissimilar distributions and a change point [12]. In TCPC, I investigate a contrastive learning technique for embedding-based CPD: Contrastive Predictive Coding (CPC). CPC is a self-supervised framework that trains an embedding model by learning to predict future latent representations from past data, known as the context [13]. This predictive structure encourages the model to capture high-level trends and patterns, forming meaningful embeddings. To detect change points, TDOST embeddings from two sequential windows are first generated and fed into a trained CPC model, where the output embeddings are then compared using the cosine similarity distance metric. A significant drop in similarity indicates a potential change point [12], allowing for detection without reliance on pretrained classifiers or labeled data.

More effective models for data segmentation can enhance data formatting, providing more consistent inputs for activity classifiers and improved predictive performance during the classification phase. To demonstrate the utility of the proposed segmentation approaches, they will be evaluated against two leading methods: Separation (SEP), a prominent statistics-based method with demonstrated effectiveness in the smart home domain [9], and Context Cosine Similarity Score (CCS), a specialized embedding-based method for HAR [8]. These experiments, measuring CPD accuracy, will showcase the superiority of the proposed techniques across diverse smart home environments. Furthermore, to demonstrate the relevance of improved data segmentation, the proposed methods will be integrated into a complete activity recognition pipeline. Within this framework, the enhanced CPD strategies will be compared to the conventional sliding window method in terms of downstream classification performance. This evaluation will demonstrate the potential of CPD to enhance the robustness of HAR systems. By tailoring segmentation techniques to the specific characteristics of HAR, the proposed pipeline introduces a more context-aware and adaptable framework for activity prediction, particularly valuable in dynamic environments where human behavior may vary unpredictably and existing segmentation strategies often fall short. Ultimately, by introducing novel paradigms for CPD in smart home sensor data, this work seeks to improve the accuracy and reliability of HAR, with broader implications for the advancement of automated smart home systems.

CHAPTER 2

LITERATURE REVIEW

This literature review explores how temporal segmentation techniques can enhance smart home sensor data processing, focusing on existing HAR approaches and the integration of modern CPD methods. The review is structured into three main sections: the first examines HAR methodologies, the second explores various segmentation techniques, and the final section investigates contrastive learning and its applicability in improving segmentation performance.

2.1 Human Activity Recognition

HAR in smart homes, a classification problem, is the process of using data from ambient sensors embedded within the environment to automatically recognize human activities [14]. HAR has been traditionally performed through a five-step activity recognition chain consisting of data collection, preprocessing, segmentation, feature extraction, and classification [15]. Research in HAR tends to focus on optimizing these individual stages, with significant emphasis on the feature extraction and classification components [3].

A well-recognized model for feature extraction and classification is DeepCASAS, introduced by Liciotti *et al.* [5]. This approach employs LSTM recurrent neural networks to model spatio-temporal smart home sensor data. Empirical evaluations demonstrate that DeepCASAS provides improvements over existing machine learning models, such as Naive Bayes (NB) and Hidden Markov Models (HMMs), particularly in handling sequential data inherent to smart home environments [5]. The key strength of the LSTM architecture exploited in the DeepCASAS approach is the ability to retain sequential information over time through cycles in the network, specialized memory cells, and input, output, and forget gates [16, 17]. Analyzing the data, DeepCASAS has been shown to outperform both conventional machine learning models and more modern Convolutional Neural Networks (CNNs) by 6–7%, verifying the improved capability of LSTM-based models in processing temporal data from smart home environments [5].

A fundamental limitation of the DeepCASAS approach, as well as other approaches in the field, is the lack of focus on transfer learning [3]. These approaches often overlook the generalizability of their pretrained models on unseen datasets, which is important given the variety of smart home environments in practice. To address this limitation while building on the success of LSTM-based models introduced by DeepCASAS, Thukral *et al.* [4] propose the TDOST framework. TDOST introduces a novel method for creating a consistent embedding space across domains by incorporating contextual information from sensor data. The

approach first converts sensor readings into textual descriptions (sentences), incorporating sensor modality and location, before transforming them into embeddings using pretrained transformers. Subsequently, these embeddings are fed into a Bi-LSTM model for training, maintaining consistency with the LSTM architecture from DeepCASAS [4, 5]. Empirical results show that TDOST outperforms DeepCASAS in transfer learning scenarios, achieving significantly higher ($> 15\%$) F1 scores. This improvement highlights the importance of embedding spaces for generalization across different environments.

Training and evaluating HAR models across the entire activity recognition chain requires substantial sensor data. Much of the current research relies on datasets from the Cognitive Assistance for Smart Apartments (CASAS) project, which developed the Smart Home in a Box (SHiB) system. While originally intended as an affordable sensor deployment kit, the CASAS project is better recognized for its comprehensive dataset, encompassing 32 annotated smart home testbeds [18, 3]. Among these, 19 represent single-resident homes, reducing the complexity of overlapping activities and aiding in more accurate model training. The public availability of the CASAS datasets makes it a central resource in HAR research. However, a key limitation remains: the relatively small scale of available data. With only 19 single-resident testbeds and limited activity annotations, training models that generalize across diverse home environments and user behaviors remains a significant challenge. Furthermore, with few change points between activities, supervised CPD approaches might struggle to generalize effectively.

2.2 Data Segmentation Techniques

As previously mentioned, segmentation is crucial for effective HAR, as it minimizes the inclusion of extraneous activities in the input data and provides a precise time frame for each recognized activity. For HAR, all applicable segmentation techniques must operate online given the continuous influx of sensor data as activities occur. A recent survey on time series segmentation by Wang *et al.* [19] formalizes key terminology, distinguishing between CPD, boundary point detection, and state detection. CPD identifies shifts in statistical properties of the data, while boundary point detection identifies semantic shifts. State detection involves assigning labels to different segments of data. Notably, CPD is a subset of boundary point detection, while state detection inherently performs boundary point detection during the labeling process. For HAR, it is sufficient to identify change points, which is accomplished through both CPD and boundary point detection. Additionally, the study mentions the relevance of Piece-wise Linear Approximations (PLAs) and Piece-wise Linear Representations (PLRs), which segment the data into linear components with respect to time, to capture trends [19, 20]. Across existing literature on temporal segmentation, modern approaches can be categorized as either unsupervised or supervised techniques according to the data availability and usage.

For unsupervised approaches, an early work on time series segmentation by Keogh *et al.* [20] identified recursive and sliding window strategies for PLR, creating segments of temporal data. The paper introduced the Sliding Window and Bottom-Up (SWAB) approach, now recognized as a clustering technique [10], which incorporates the online nature of sliding window approaches with the accuracy of bottom-up approaches. SWAB uses a buffer and a simple combination of techniques to accomplish its objective. Another work by Plotz *et al.* [11] introduced the Snip-Snap technique for automated CPD, using projections created through HMMs of a snip window onto a snap window to detect changes in data. Modern unsupervised approaches, like the likelihood ratio method discussed in Aminikhanghahi and Cook [10], employ a similar strategy to Snip-Snap, using probabilistic modeling—typically with a multivariate Gaussian—to model data distributions across sequential windows. In Aminikhanghahi and Cook [10], likelihood ratio methods like Relative Unconstrained Least-Squares Importance Fitting (RuLSIF) show the best performance, with a high Area Under Curve (AUC) scores on the Receiver Operating Characteristic (ROC) curve, which measures the ratio of true positives to false positives. Subspace modeling is a close contender in terms of performance. The Fast Low-Cost Online Semantic Segmentation (FLOSS) method introduced by Gharghabi *et al.* [21], was recognized for its low cost and highly efficient performance in boundary point detection by Wang *et al.* [19]. FLOSS is a profile-based method that uses a data structure encoding the distances between subsequences of a time series, titled a matrix profile, to generate a Corrected Area Curve (CAC). The local minima of the CAC represents boundary points. SEP was introduced as a kernel-based successor to RuLSIF and other density-ratio approaches to CPD. It maintains two adjacent sliding windows and directly estimates the probability-density ratio between them via a kernel expansion (typically using Gaussian kernels). The ratio parameters are obtained in closed form through unconstrained least-squares importance fitting, which keeps the computation lightweight. From this fitted ratio, SEP derives a bounded separation-distance score; when the score forms a local peak that exceeds a preset threshold, the method labels a change point. Because separation distance emphasizes the largest drop in probability mass, unlike the α -relative Pearson divergence used by RuLSIF, SEP is more sensitive to subtle or rare-event changes and achieves better detection performance without requiring additional tuning [9]. Aminikhanghahi and Cook [10] propose State detection methods like Efficient and Unsupervised State Detection (E2USD) and Streamscope can be applied to boundary detection inherently, but lack the foundational focus in design.

Supervised approaches recognized in Aminikhanghahi and Cook [10] use classifiers such as decision trees, NB, Support Vector Machines (SVMs), and HMMs to classify change points either through multi-class classification or binary classification. Such approaches require extensive training data that captures all types of change points available in the environment, which might be impractical in the HAR domain given the lack of labeled datasets and domain variety [10, 3]. An interesting supervised approach assigns binary labels to

two windows surrounding a change point, using virtual classifiers to identify a statistical difference based on the confidence of the classifier’s decision-making [10].

While most HAR implementations use the fixed sliding window approach for data segmentation, unsupervised and supervised CPD techniques have been developed or adapted specifically for HAR. An effective CPD algorithm for HAR should achieve a true positive rate close to 100%, as missing key activity windows could compromise downstream classification. A higher false positive rate, however, is tolerable, as incorrect boundaries can still be accurately classified and conjoined into larger data segments, similar to SWAB, minimizing their impact. Aminikhanghahi *et al.* [9] adapt unsupervised statistics-based techniques from Aminikhanghahi and Cook [10] to HAR, but demonstrate a low exact CPD true positive rate of 40%, requiring a significant buffer to achieve a 90% recall. Najeh *et al.* [7] and Wan *et al.* [6] both employ correlation coefficients between sequential sensor reading encodings to identify change points, but report high error rates in CPD due to variability in activity performance. Embedding-based approaches, such as CCS introduced in Bermejo *et al.* [8], train models to encode sensor data into embeddings, comparing distances between windows to detect change points. While the approach shows promise, it exhibits inconsistency between domains, achieving a true positive rate of only 46% in worst-case home layouts. However, the paper highlights potential for improved recall with better embedding models. The proposed embedding model in the paper is trained by optimizing the likelihood of a sensor reading given its context window; alternatively, training the embedding model to maximize the separability of embeddings at change points in the latent space could yield better results.

2.3 Contrastive Learning

Modern segmentation techniques for temporal data tend to rely on comparing two windows of data to detect change points. This approach aligns naturally with contrastive learning, a method typically used in self-supervised learning, where data segments are compared in pairs. In contrastive learning, a data entry is selected, and the remaining entries are categorized as either positive or negative samples, with the goal of projecting positive samples closer together and negative samples farther apart in the embedding space [22, 23]. Contrastive learning for embedding-based temporal segmentation in smart home sensor data offers the potential to effectively compare sequential windows of data, identifying meaningful differences in their resulting embeddings which may correspond to change points.

Contrastive learning architectures are typically categorized into four categories: end-to-end learning, memory banks, momentum encoders, and clustering-based approaches [23]. End-to-end learning uses a query and key encoder, where the query encoder matches positive samples and the key encoder differentiates

between positive and negative samples. Memory banks and momentum encoders are both improvements on the memory limitations with negative sampling, reducing the need for large batch sizes. Lastly, clustering-based methods abstract from the previous architectures by creating meaningful groups of data for implicit differentiation, without requiring explicit positive and negative pairs. In these methods, augmented samples are projected toward learned cluster centers, with different clusters representing distinct categories of samples. The learned representations using the advanced clustering addresses a limitation in existing end-to-end approaches where negative samples are incorrectly identified and contrasted.

One of the most well-known end-to-end contrastive learning models is the Simple Framework for Contrastive Learning (SimCLR) introduced by Chen *et al.* [24]. However, SimCLR’s reliance on large batches with numerous negative samples makes it less practical for the HAR domain, where labeled data is often sparse. To overcome this limitation, Grill *et al.* [25] introduced Bootstrap Your Own Latent (BYOL), an approach that mitigates the need for negative samples. BYOL uses two networks, an online and a target network, that share the same architecture consisting of an encoder, projector, and predictor. The online network is trained to match the representations generated by the target network, and the target network is updated occasionally with the exponential moving average of the online network’s weights. The asymmetry between the networks prevents the model from collapsing to a trivial solution. Building on BYOL, Zbontar *et al.* [26] developed the Barlow Twins (BT) approach, which simplifies the architecture by using a redundancy-reduction principle. Instead of explicit positive and negative pairs, BT optimizes the cross-correlation matrix between two positive samples, attempting to align it with the identity matrix. Empirical evaluations show that BT performs comparably to BYOL, particularly excelling in high-dimensional representation spaces and low training batch sizes. However, in most practical experiments with moderate batch sizes and reasonable representation dimensions, BYOL continues to outperform BT in terms of accuracy [26].

While contrastive learning is traditionally employed as an unsupervised technique, operating on unlabeled datasets, supervised contrastive learning has been introduced to leverage labeled data for enhanced embeddings [27]. In contrast to techniques like SimCLR, which assume that each data point represents a unique class and treat all other points as negative samples [24], supervised contrastive learning accounts for scenarios where multiple data points belong to the same class. This is particularly relevant for HAR datasets like CASAS, where activities are finite and repeated many times [18], leading to confusion in the latent space of unsupervised models. By utilizing labeled datasets and introducing a Supervised Contrastive (SupCon) loss function, supervised contrastive learning explicitly projects data points of the same class together while pushing points from different classes apart, outperforming state-of-the-art unsupervised methods. Mainly, while approaches like SimCLR are designed for feature extraction for downstream tasks such as classification, supervised contrastive learning can be directly applied for embedding-based CPD as proposed by Bermejo *et*

al. [8].

CPC [13] offers an alternative perspective within the contrastive learning paradigm, particularly suited for sequential data. Unlike SimCLR, BYOL, BT, and SupCon, which rely on augmented views of the same instance to form the embedding space, CPC capitalizes on the inherent temporal structure of its input. Specifically, CPC employs an encoder coupled with an autoregressive model to predict future latent representations based on past context. The model is trained to distinguish the true future representation from a set of randomly sampled negatives through the contrastive Information Noise-Contrastive Estimation (InfoNCE) loss function. By necessitating the identification of meaningful temporal features, CPC generates robust representations that capture important characteristics of the data. The predictive framework is especially promising for domains like HAR, where sensor data exhibits notable temporal dependencies. Moreover, while other approaches often depend on an abundance of different labeled segments for adequate training, CPC leverages the temporal nature of HAR data to construct its embedding space, making it an ideal option for encoding. CPC embeddings have been shown to improve HAR performance compared to alternative encoding approaches [28]. Identifying the potential of CPC for CPD, Deldari *et al.* [29] introduce Time Series Change Point Detection Based On Contrastive Predictive Coding (TS-CP²), a contrastive framework that learns representations by distinguishing between temporally adjacent and non-adjacent segments. The similarity scores of these learned representations over time can be used to detect distributional shifts and indicate a likely change. Although this approach has shown improvements over leading statistics-based CPD methods such as RuLSIF, as previously showcased by Thukral *et al.* [4], it may face challenges in generalizing across diverse smart home environments.

CHAPTER 3

FRAMEWORKS & METHODS

This project proposes two CPD algorithms to improve segmentation for HAR: TAD, a supervised method using a pretrained HAR classifier, and TCPC, an unsupervised method using CPC embeddings. This section details the materials and methodology for training and deploying both approaches.

3.1 Materials

Since this work is primarily software-based, materials include digital libraries and smart home datasets. Table 3.1 summarizes each material, its description, intended use, and source. All software packages are Python-compatible to facilitate model development and testing given the language’s prevalence in modern machine learning.

Table 3.1: Summary of the Materials Used in the Project.

Material	Description	Application	Source
PyTorch	Open-source Python library for deep learning	Model building and training (HAR classifier, CPC feature extractor)	Paszke <i>et al.</i> [30]
NumPy	Open-source library for efficient array operations in Python	Sensor data preprocessing and manipulation	Harris <i>et al.</i> [31]
SciPy	Open-source Python library with additional numerical computing functions	Identifying local maxima for CPD	Virtanen <i>et al.</i> [32]
Scikit-learn	Open-source Python machine learning library with data analysis utilities	Data preprocessing and train-validation-test splits	Pedregosa <i>et al.</i> [33]
CASAS Smart Home Datasets	Manually annotated sensor data from smart homes	Data source for training and experimentation for the proposed approaches	Cook <i>et al.</i> [18]

3.2 Methodology

The methodology introduces the two approaches to change point detection, TAD and TCPC, separately, with its model design, training procedure, and deployment process. Unless otherwise stated, the models listed are implemented using `PyTorch 2.6.0`.

3.2.1 Approach 1: Supervised Learning (TAD)

The supervised approach, TAD, depicted in Figure 3.1, employs the state-of-the-art pretrained HAR classifier [5, 4] with the Snip-Snap technique [11]. The Snip-Snap method uses the classifier’s predicted activity distributions on two consecutive windows—Snip (past) and Snap (future)—to detect differences that indicate a change point. High dissimilarity between these distributions indicates a likely change in activity.

This Snip-Snap technique is preferred over conventional binary change point classifiers for two main reasons. Firstly, it addresses class imbalance. In continuous HAR data streams, change points are relatively sparse, so most windows belong to the dominant “no change” class. This either biases the binary detector towards the “no change” class, or forces aggressive downsampling of negatives, both of which hurt performance [10]. The second reason relates to combinatorial data scarcity. A change point can be defined by the pair of activities it separates; with a set of A activities, there are $|A|(|A| - 1)$ possible types of change points. Collecting enough examples for each pair might not be plausible, so a binary detector can struggle with unseen or underrepresented transitions. The Snip-Snap approach avoids both problems by repurposing the abundant activity labels. It compares two activity-probability vectors and flags a boundary when they diverge. The learning task thus scales linearly with the number of activities, not quadratically with their pairs, leading to simpler training and better generalization.

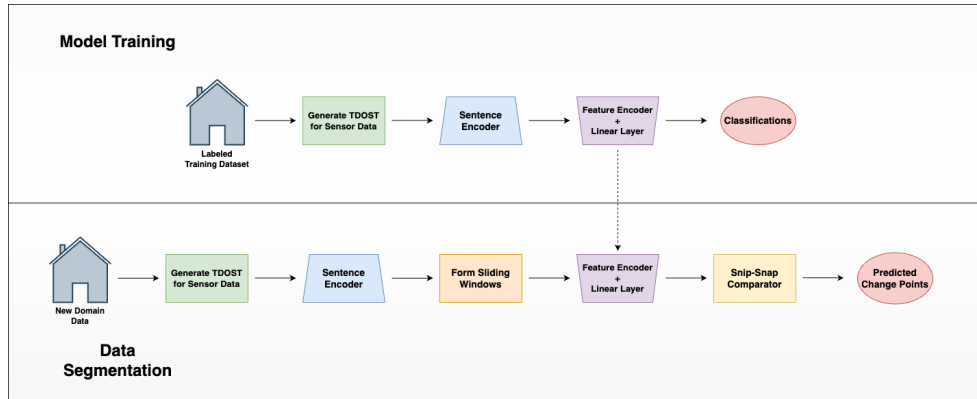


Figure 3.1: The supervised approach to data segmentation

HAR Classifier Architecture

The HAR classifier is a two-layer Bidirectional Long Short-Term Memory (BiLSTM) sequence encoder followed by a single fully connected layer and a softmax activation that outputs class posterior probabilities. To motivate the design, because smart home sensor data inherently contain complex temporal dependencies, bidirectionality and its ability to exploit both past and future temporal contexts offers an advantage in mod-

eling such dependencies [34]. Dropout ($p = 0.2$) is applied after each LSTM layer to curb overfitting and improve generalization [35]. This template has delivered state-of-the-art activity recognition performance on several CASAS datasets [5, 4].

Data Preparation

Before training, activity labels are unified across datasets using the label mappings from Liciotti *et al.* [5]. For example, all variants of SLEEPING are merged into a unified canonical label. Each sensor event is then converted to its TDOST textual description and embedded with the Sentence-Transformers `all-distilroberta-v1` model as prescribed in Thukral *et al.* [4]. The resulting 768-dimensional embeddings are grouped by activity segment and either padded or truncated to a fixed context window¹ of $k = 5$. The resulting sequence $x_{1:k}$ is passed to the BiLSTM.

Training Procedure

A three-fold chronological split is adopted for dividing each dataset. In each fold, 66% of the data forms the training-validation pool, while the remaining 34% is used for testing. Within the pool, a further 20% is used for validation². A fixed random seed of 42 helps reproducibility and aligns with the protocol of Thukral *et al.* [4].

Unless otherwise stated, the hyperparameter mappings from Thukral *et al.* [4] are employed. Models are trained for 75 epochs using the Adam optimizer, and a step learning-rate scheduler with a step size of 10 and a decay factor of $\gamma = 0.8$ for smooth convergence.

Given a labeled sequence of (x, y) , the model parameters θ are learned by minimizing the categorical cross-entropy loss

$$\mathcal{L}(\theta; x, y) = - \sum_{a \in A} y_a \log(\hat{y}_a(x; \theta)),$$

where A is the activity set, y_a is the one-hot ground truth, and \hat{y}_a is the predicted probability for class a . The cross-entropy loss measures the dissimilarity between a predicted probability distribution and the true class distribution, so minimizing this loss ensures the model assigns a higher probability to the correct class [36]. After each epoch, the weighted F1 score is computed on the validation split, and the checkpoint with the highest score is retained for downstream deployment.

¹This context window was empirically observed to achieve the best change point performance, as depicted in Table A.1.

²The train/validation/test proportions are approximately 53%/13%/34%.

Deployment

At inference time, the classifier must label and then segment an unbounded stream of sensor events. Each incoming sensor trigger is first mapped to its TDOST representations, embedded with the same sentence transformer used in training, and pushed onto a FIFO buffer.

Every s sensor triggers³, two fixed-length windows of size k are materialized:

$$\mathbf{x}_{t-k:t} \quad (\text{Snip, past context}), \quad \mathbf{x}_{t:t+k} \quad (\text{Snap, future context}).$$

A slight buffer is employed during deployment to ensure both windows are well-defined.

The classifier g_{class} produces posterior probability vectors

$$\hat{\mathbf{y}}_{\text{snip}}(t) = g_{\text{class}}(\mathbf{x}_{t-k:t}), \quad \hat{\mathbf{y}}_{\text{snap}}(t) = g_{\text{class}}(\mathbf{x}_{t:t+k}).$$

To identify activity boundaries, the paired posteriors are collapsed into a scalar divergence curve. Cosine dissimilarity—one minus the cosine similarity—is used because of its improved performance⁴:

$$\text{DIS}(t) = 1 - \cos(\hat{\mathbf{y}}_{\text{snip}}(t), \hat{\mathbf{y}}_{\text{snap}}(t)) = 1 - \frac{\hat{\mathbf{y}}_{\text{snip}}(t)^\top \hat{\mathbf{y}}_{\text{snap}}(t)}{\|\hat{\mathbf{y}}_{\text{snip}}(t)\|_2 \|\hat{\mathbf{y}}_{\text{snap}}(t)\|_2}. \quad (3.1)$$

Since both posterior vectors are probability distributions, it is found that $\text{DIS}(t) \in [0, 1]$, where a value of 0 indicates the two windows predict exactly the same activity mixture, while values approaching 1 indicate the windows place probability mass on disjoint sets of activities.

Local maxima of the dissimilarity curve are labeled as change points, marking instances where the predicted activity distribution shifts drastically. The contiguous intervals between successive change points define variable-length segments which are fed to downstream activity classification modules for fine-grained labeling or additional adaptation.

3.2.2 Approach 2: Self-supervised Learning (TCPC)

In contrast to the supervised classifier-based TAD method, the unsupervised TCPC approach, depicted in Figure 3.2, uses CPC to learn feature representations without activity labels. The Snip-Snap segmentation procedure is then applied directly to these output features rather than to classifier outputs.

³A default stride of $s = 1$ is used to avoid missing any change points.

⁴The cosine dissimilarity metric was chosen over the dot product and Jensen-Shannon divergence based on an empirical evaluation of their performance in change point detection, as shown in Table A.1.

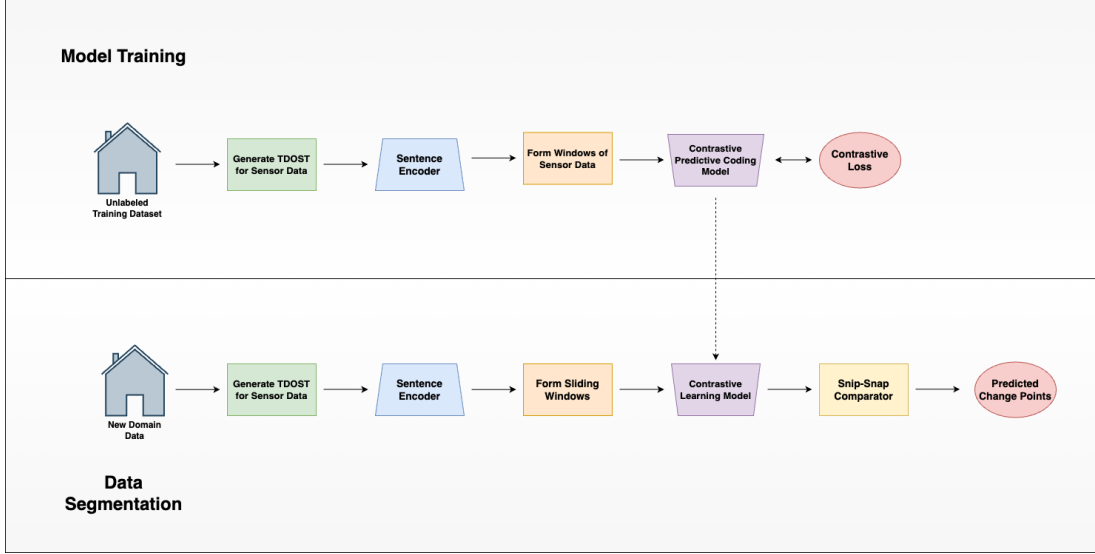


Figure 3.2: The unsupervised approach to data segmentation

CPC Architecture

A standard CPC model consists of two components: an encoder and an autoregressive network. For HAR, this template can be adopted following Haresamudram *et al.* [28]. Input embeddings are fed into a convolutional encoder consisting of three convolutional blocks, which downsample the feature dimension. The encoded sequence $g_{\text{ENC}}(\cdot)$ is then passed to a two layer Gated Recurrent Unit (GRU) with a dropout of $p = 0.2$, which serves as the autoregressive context model. Finally, a list of trainable linear weight matrices $W_{i=1}^k$ projects the context representation c_t k steps into the future.

Training Procedure

Training is performed without annotations on a source CASAS dataset. Each sensor event is first mapped to its TDOST representation and embedded using a frozen Sentence-Transformers `all-distilroberta-v1` model. A context window length of $k = 10$ was used⁵, forming paired Snip ($x_{t-k:t}$) and Snap ($x_{t:t+k}$) sequences that are fed sequentially through g_{ENC} and the GRU.

The model is optimized using the Adam optimizer with an initial learning rate of 0.001 and a step learning-rate scheduler with a step size of 25 and a decay factor of $\gamma = 0.8$. A batch size of 64 was used, and the model was trained for 120 epochs. This hyperparameter selection mirrors the protocol of Haresamudram *et al.* [28], where a thorough empirical analysis was conducted.

⁵The optimal CPC hyperparameters were experimentally determined through a grid search depicted in Table A.3.

Following Oord *et al.* [13] and Haresamudram *et al.* [28], the objective is the InfoNCE loss

$$\mathcal{L}(\theta; x) = -E_X \left[\log \frac{f_k(x_{t:t+k}, x_{t-k:t})}{\sum_{x_{j:j+k} \in X} f_k(x_{j:j+k}, x_{j-k:j})} \right], \quad (3.2)$$

where the predictive score f_k is instantiated as

$$f_k(x_{t:t+k}, x_{t-k:t}) = \exp \left(\frac{1}{t} \sum_{i=k+1}^{t+k} x_i^T W_{i-k} g_{\text{GRU}}(g_{\text{ENC}}(x_{t-k:t})) \right). \quad (3.3)$$

Minimizing (3.2) encourages f_k to assign higher scores to correct future predictions rather than to random negative samples, thereby forcing both the encoder and GRU to capture predictive structure in the data. To better illustrate, let $c_t(x_{t-k:t}) = g_{\text{GRU}}(g_{\text{ENC}}(x_{t-k:t}))$ represent the context vector produced by the CPC model on input $x_{t-k:t}$. This optimization causes f_k to approximate the following density ratio [13]:

$$f_k(x_{t:t+k}, x_{t-k:t}) \propto \frac{p(x_{t+k} | c_t)}{p(x_{t+k})} = \frac{p(x_{t+k}, c_t)}{p(c_t)p(x_{t+k})}.$$

A high f_k indicates that the joint distribution of the context and future value is significantly larger than the products of their marginals, reflecting a stronger correlation between them. By optimizing for correct future predictions, the GRU and encoder are forced to learn valuable feature representations of the data [13].

The model achieving the lowest validation InfoNCE loss is checkpointed and used for deployment.

Deployment

During deployment, incoming sensor streams are converted to TDOST embeddings and segmented online by the Snip-Snap procedure. For each sliding window, Snip ($x_{t-k:t}$) and Snap ($x_{t:t+k}$) representations are computed:

$$c_{\text{snip}} = g_{\text{GRU}}(g_{\text{ENC}}(x_{t-k:t})), \quad c_{\text{snap}} = g_{\text{GRU}}(g_{\text{ENC}}(x_{t:t+k}))$$

Cosine dissimilarity, the primary metric for CPD in embedding-based approaches [8], is calculated using

$$\text{DIS} = 1 - \cos(c_{\text{snip}}, c_{\text{snap}}) = 1 - \frac{c_{\text{snip}}^\top c_{\text{snap}}}{\|c_{\text{snip}}\|_2 \|c_{\text{snap}}\|_2}. \quad (3.4)$$

Local maxima of DIS correspond to moments where the semantic content embedded in the context vectors changes abruptly. Because these boundaries are detected directly from representation dynamics, the method remains robust to mid-activity variation and does not require predefined activity labels.

CHAPTER 4

RESULTS

In this section, I present and discuss the results of my experiments. These results demonstrate that the proposed TAD and TCPC methods for CPD in smart home sensor data perform as well as or better than existing prominent statistical and embedding-based methods. For training and testing, the Aruba, Milan, Kyoto, and Cairo CASAS datasets were selected, which have been utilized in the evaluation of TDOST and are widely referenced in HAR literature [4, 5, 7].

4.1 Change Point Detection and Transferability

This section quantifies the effectiveness of the proposed TAD and TCPC frameworks for CPD, and analyses their ability to transfer to unseen smart-home environments. Their performance is contrasted against two well-established baselines in recent HAR literature—the statistics-based SEP method and the embedding-based CCS approach [8, 9].

SEP extends the popular RuLSIF method by measuring the separation distance between successive windows, with larger probability ratio discrepancies signaling a change point [9]. The implementation provided follows the authors’ recommendations and employs a Gaussian kernel. In contrast, CCS generates latent embeddings by maximizing the likelihood of a sensor reading given its encoded temporal context. High cosine dissimilarity between successive embeddings indicates a change point [8].

To emphasize the transferability of each method, TAD and TCPC are trained exclusively on the training portion of the Aruba dataset and are never fine-tuned on the target domains (Milan, Kyoto, and Cairo). Results on Aruba are reported only for its isolated test split. Consequently, any performance differences can be attributed to each model’s ability to identify change points across various homes, layouts, and sensor modalities.

Consistent with prior work, the True Positive Rate (TPR), False Positive Rate (FPR), and the geometric mean (G-mean) of TPR and True Negative Rate (TNR) were used as success metrics [9]. They are defined as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad \text{G-mean} = \sqrt{\text{TPR} \cdot (1 - \text{FPR})}$$

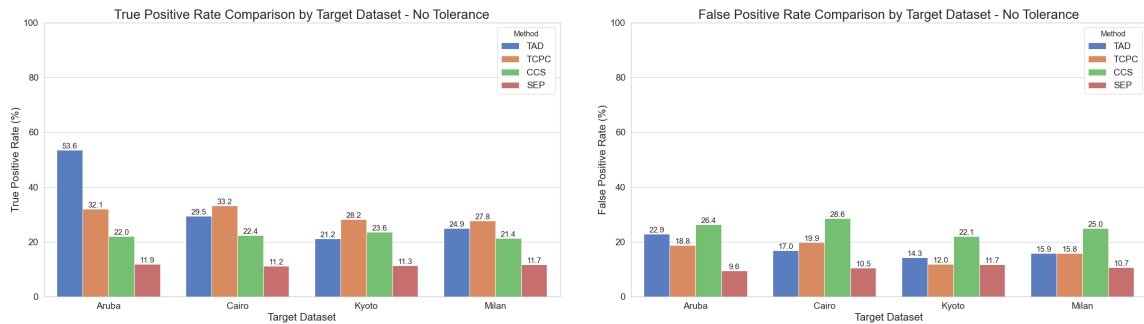
where TP represents the number of correctly identified change points (true positives), FN denotes the number of change points missed (false negatives), and FP indicates the number of non-change points incorrectly

classified as change points (false positives).

A high TPR, or recall, indicates that actual change points were rarely missed, whereas a high FPR reflects oversegmentation, where non-change points were frequently misclassified as change points. A good CPD algorithm should maximize TPR while minimizing FPR. Although recall is generally prioritized over precision for downstream activity recognition tasks, since oversegmentation can be merged later with an effective classifier, the G-mean balances both aspects and therefore serves as our primary success metric.

Consistent with other works [9, 8, 10], two evaluation protocols were adopted:

1. **Exact Matching:** A prediction is counted as correct only when its timestamp coincides exactly with a ground-truth change point.
2. **Tolerant Matching:** A tolerance window of ± 5 s is allowed, accounting for annotation uncertainty and minor, unimportant inaccuracies in the identification of change points.



(a) Exact Change Point Detection True Positive Rate

(b) Exact Change Point Detection False Positive Rate

Figure 4.1: Exact Change Point Detection TPR and FPR

4.1.1 Exact Change Point Detection

Figures 4.1–4.2 summarize the findings of the exact change point detection experiments. Several key observations stand out:

- **High Recall:** Across all four target homes, TAD obtained the highest TPR on its source domain (Aruba), while TCPC achieved the highest recall on every unseen environment. This is evidence that the self-supervised approach, using TDOST and CPC, learns spatial- and sensor-agnostic representations with good performance in change point detection. Additionally, these results showcase the strength of the supervised model on its source dataset, as TAD significantly outperformed the remaining techniques.

- **Mitigating Oversegmentation:** Despite their superior recall, TAD and TCPC keep FPR markedly below CCS and comparable to SEP, indicating the proposed models' learning of robust decision boundaries rather than simple guessing.
- **Geometric Mean:** The G-mean confirms the qualitative analysis: TCPC ranks first on three of the four smart homes, with TAD retaining a narrow edge on Aruba owing to its supervised advantage.

These results demonstrate that the proposed methods significantly outperform existing baselines under the strictest CPD evaluation protocol.

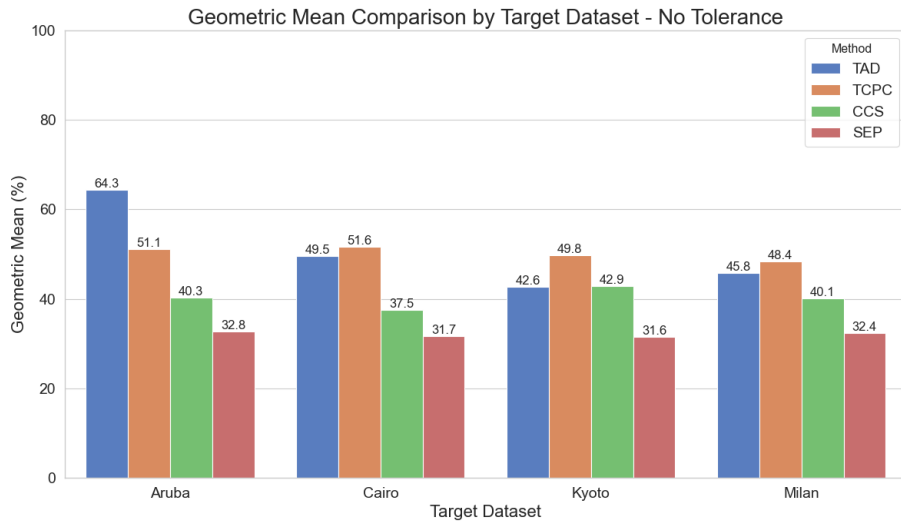


Figure 4.2: Exact Change Point Detection Geometric Mean

4.1.2 Tolerant Change Point Detection

Allowing a ± 5 s tolerance naturally inflates recall and, to a less extent, reduces FPR.

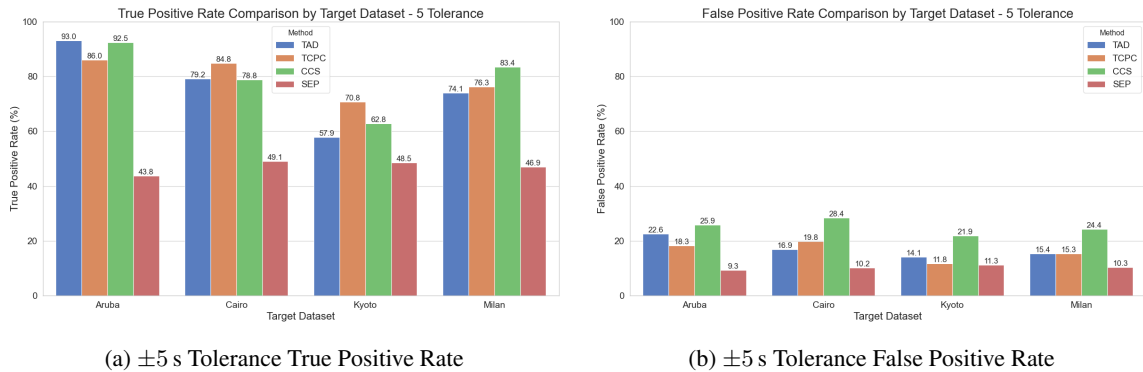


Figure 4.3: ± 5 s Tolerance Change Point Detection TPR and FPR

Figures 4.3–4.4 reveal that methods’ relative rankings remains largely unchanged, with TCPC dominating on all target homes, while TAD retains an edge on the Aruba source dataset. Interestingly, the performance gap shrinks with the introduction of tolerance, a result of the class imbalance between change points and non-change points present in the datasets.

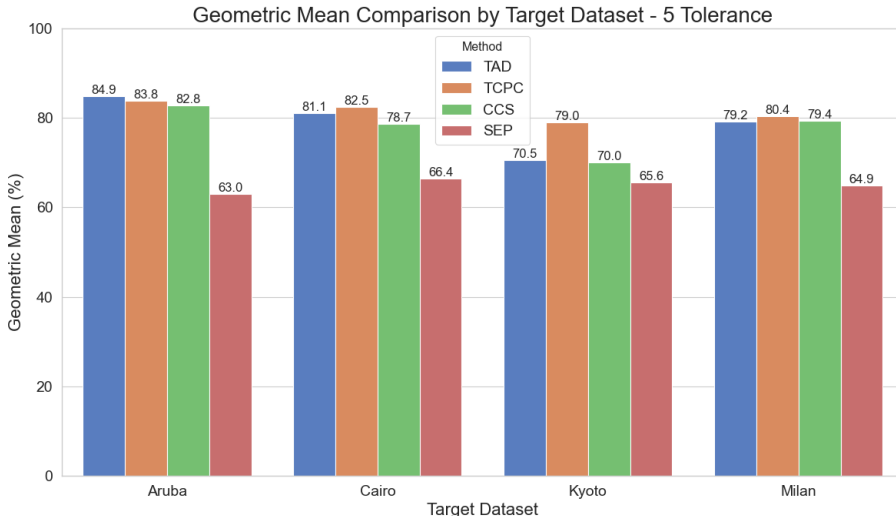


Figure 4.4: ± 5 s Tolerance Change Point Detection Geometric Mean

Table 4.1: Average Geometric Mean for Each Method Across the Target Datasets

Method	Average G-Mean
TAD	78.93
TCPC	81.43
CCS	77.73
SEP	64.98

Overall, TCPC outperforms the other methods by 2.50%-16.45% in average G-mean, confirming that CPC and TDOST encodings capture domain-invariant activity dynamics that can be effectively used for change point detection. Remarkably, the supervised TAD method also outperforms the unsupervised statistics and embedding-based CPD benchmarks in average G-mean, showcasing the ability for TDOST to assist supervised models in generalizing to new environments.

4.2 Downstream Classification

This experiment quantifies how our improved segmentation quality impacts HAR in smart homes. The naive sliding window technique, a common approach in HAR deployment [3, 7], is used as a benchmark for comparison against our two novel CPD frameworks, on the Aruba CASAS dataset.

All segments are fed to the BiLSTM classifier detailed in Section 3.2.1, previously shown to perform strongly on presegmented data [4]. Each of the segmentation approaches can be described as follows:

- **Sliding Window:** Fixed windows of $k = 5$ events¹ with a stride of $s = k/2$. Overlapping predictions are merged by majority vote.
- **TAD:** Change points are detected using an activity classifier through the procedure listed in Section 3.2.1.
- **TCPC:** Change points are detected using a CPC model through the procedure listed in Section 3.2.2.

The Macro F1 score is measured, an unweighted mean across the F1 score of all activity classes. The F1 score is the harmonic mean of precision and recall, and can be calculated as

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The Macro F1 score penalizes trivial solutions that overpredict the dominant OTHER class, making it a more reliable measure of performance in deployment scenarios with the class imbalance observed in Aruba [37].

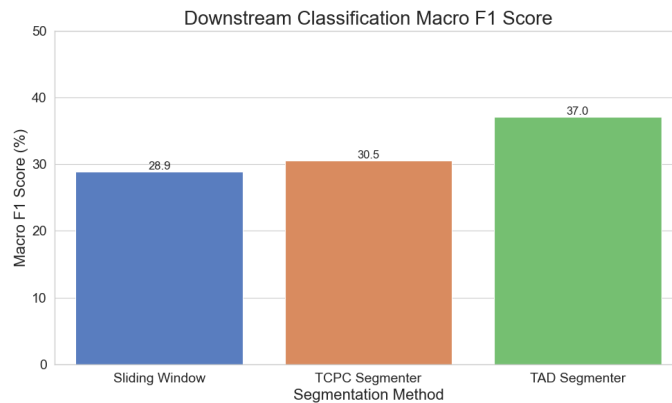


Figure 4.5: Classification Macro F1 Score

Figure 4.5 illustrates the findings. Both TCPC and TAD methods outperform the sliding window baseline, highlighting the importance of semantically meaningful segmentation. TAD yields the largest gain, boosting Macro F1 by 8.18% relative to the baseline. This mirrors its superior exact CPD recall reported in Section 4.1.1. TCPC also improves performance, though its benefit is moderated by the lower recall of its unsupervised change point estimates. These findings confirm that HAR performance depends not only on classifier design, but equally on how sensor streams are partitioned.

¹The window length was selected by considering the upper quartile gap between sensor readings in the validation sets for TAD and TCPC.

CHAPTER 5

DISCUSSION

This section reflects on the research contributions of this project, situates them within the broader HAR field, and outlines potential directions for future work. In summary, I introduced two deep learning-based CPD methods, TAD and TCPC, and demonstrated how their integration of the Snip-Snap method and TDOST achieves state-of-the-art segmentation performance on smart home sensor data. Furthermore, I illustrated how these segmentation techniques can enhance downstream HAR performance, highlighting their practical impact and real-world applications. I now go beyond quantitative improvements, and consider the broader implications, limitations, and opportunities for future research that emerge from these findings.

5.1 Limitations

While our experiments demonstrate clear improvements in TPR, G-mean, and downstream classification Macro F1, several limitations remain. First, the proposed models perform well when a tolerance window is applied; however, they often fail to detect precise change point instances. These inaccuracies propagate to HAR, where minor temporal offsets can confuse the classifier and degrade its performance. Second, although TAD and TCPC are more lightweight than many Transformer-based approaches, the classifier and CPC model still demand considerable computational resources. This is particularly problematic in HAR applications, where deployment typically occurs on resource-constrained devices. As such, both memory footprint and inference latency warrant further investigation. Lastly, our techniques were evaluated exclusively on the CASAS dataset, which includes a limited range of sensor modalities and types. Moreover, recent studies have begun to highlight potential issues within the CASAS dataset, which may undermine the generalizability of our findings [38].

5.2 Future Work

To translate these algorithms from the lab to real smart homes, there are various promising future research directions. Firstly, alternative contrastive or classification models for TCPC and TAD could be explored. Given the strong generalization performance of TAD on unseen datasets, methods such as SupCon may further enhance CPD effectiveness. Additionally, leveraging the representational capacity of Transformer architectures, paired with parameter-efficient techniques like quantization and model distillation, could yield performance gains while maintaining computational feasibility.

Secondly, addressing the limitations of the CASAS datasets is a crucial problem to address. For example, adopting improved annotation protocols with finer granularity may help mitigate oversegmentation issues. Additionally, standardizing annotation procedures across datasets could also enhance consistency and reproducibility of results.

Thirdly, further investigation into the computational efficiency of the proposed change point methods is warranted. Evaluating runtime and memory usage during both training and inference would provide valuable insights into their suitability for resource-constrained deployment environments, as is expected in HAR.

Finally, due to the prevalence of variable-length sequences in HAR, future work should explore training HAR classifiers using more diverse input lengths. Incorporating variable-length inputs with purposeful padding may more accurately reflect deployment conditions and improve model robustness.

5.3 Significance

The algorithms proposed in this research offer significant contributions to the HAR field, particularly due to their computational efficiency, superior performance, and real-time applicability in data segmentation across diverse smart home environments.

Section 4.2 showed that our improved segmentation techniques boost classification Macro F1 by more than 8% over traditional methods, directly reducing misclassified activities in real deployments. By incorporating TDOST, the approaches are layout-agnostic and require minimal configuration for usability. Thus, the presented methods are capable of advancing diverse smart home systems, helping facilitate more reliable and accurate activity classification.

Ultimately, this research introduces two novel, effective, and computationally feasible approaches for CPD in smart home sensor data. Despite the highlighted limitations, empirical results offer valuable guidance for future research toward improving HAR robustness.

CHAPTER 6

CONCLUSION

This thesis aimed to address two primary research questions: first, whether existing data segmentation techniques in HAR can be improved; and second, whether these techniques significantly impact HAR performance. The results of the two proposed CPD methods, TAD and TCPC, provide affirmative answers to both questions.

To begin, I employed TDOST to standardize sensor inputs across varying smart home layouts. The segmentation problem was then reformulated in terms of representation divergence. By comparing either activity-probability distributions (TAD) or CPC-generated context vectors (TCPC) across adjacent time windows, scalar dissimilarity curves whose peaks served as indicators of change points are derived.

Evaluations conducted across four distinct CASAS smart homes demonstrated that TAD achieved the highest exact-match recall within its source domain, while TCPC exhibited superior generalization to unseen layouts. Both approaches consistently outperformed statistical and embedding-based CPD benchmarks in terms of geometric mean performance. Importantly, these improvements were realized without any parameter tuning during domain transfer, indicating that the learned representations capture underlying activity dynamics rather than layout-specific features, and are thus capable of generalizing effectively.

Beyond enhancing change point detection, I assessed the impact of these methods on downstream classification performance, in comparison to the standard sliding window approach. While TCPC led to a modest 1.66% improvement in Macro F1 score, TAD achieved a substantial 8.18% increase, resulting in notably fewer missed or spurious activity labels. These findings confirm that the proposed methodologies not only surpass existing segmentation techniques, but also contribute meaningful gains to overall HAR accuracy, even in conjunction with unoptimized classifiers.

In conclusion, this work addresses the underrepresented challenge of data segmentation in HAR by presenting two practical, layout-agnostic, and representation-driven CPD solutions. By demonstrating both improved accuracy and enhanced generalizability, this thesis lays the groundwork for more adaptive and intelligent smart home systems, ones that can better understand and respond to the behaviors of their occupants, thereby enriching daily living for millions worldwide.

Appendices

Appendix A

Change-Point Detection: Metric Selection and Hyperparameter Optimization

I empirically justify the design decisions made in the TAD and TCPC implementations. Specifically, for TAD, I focus on the choice of the dissimilarity metric and the context window length. For TCPC, I investigate hyperparameters during CPC training that most strongly affect downstream change point detection.

Table A.1 evaluates a TAD model trained on the Aruba dataset and evaluated on its validation set, comparing three similarity measures—dot product, cosine similarity, and Jensen–Shannon divergence—across context windows of 5, 10, 20, and 50 frames. Accuracy is reported for both exact change point matches and detections within a pragmatic ± 5 s tolerance. Under the tolerant criterion, cosine similarity with a five-frame window yields the highest geometric mean (81.41 %), and this setting is therefore adopted for all subsequent TAD experiments.

Tables A.2 and A.3 depict a grid search over the CPC hyperparameters most influential to TCPC: context length, the number of future steps predicted during training, and the optimizer learning rate. Each configuration is trained on the Aruba training split and evaluated on the validation split under both exact and ± 5 s tolerance conditions. The tolerant evaluation identifies a 10-frame context, five future predictions, and a learning rate of 0.001 as optimal, achieving a true-positive rate of 0.905, a false-positive rate of 0.243, and a geometric mean of 0.820. These hyper-parameters are used in all remaining TCPC experiments.

Table A.1: TAD Change Point Detection Accuracy Across Similarity Metrics and Context Window Sizes (Exact vs. ± 5 s Tolerant) Evaluated on the Aruba Dataset.

Window	Similarity	Metric	Exact CPD	With Tolerance
5	Dot	TPR	56.13	95.17
		FPR	30.98	30.68
		G-Mean	62.25	81.22
	Cosine	TPR	56.27	95.28
		FPR	30.74	30.44
		G-Mean	62.43	81.41
	JSD	TPR	55.63	95.12
		FPR	30.97	30.66
		G-Mean	61.97	81.21
10	Dot	TPR	60.97	94.96
		FPR	34.03	33.77
		G-Mean	63.42	79.30
	Cosine	TPR	61.10	94.89
		FPR	33.46	33.20
		G-Mean	63.76	79.61
	JSD	TPR	59.65	94.96
		FPR	32.44	32.17
		G-Mean	63.48	80.25
20	Dot	TPR	59.90	95.03
		FPR	34.04	33.77
		G-Mean	62.86	79.33
	Cosine	TPR	60.01	94.89
		FPR	33.54	33.27
		G-Mean	63.16	79.61
	JSD	TPR	58.29	94.61
		FPR	33.31	33.04
		G-Mean	62.35	79.60
50	Dot	TPR	54.66	95.10
		FPR	33.32	33.01
		G-Mean	60.37	79.81
	Cosine	TPR	54.87	95.10
		FPR	33.11	32.80
		G-Mean	60.58	79.94
	JSD	TPR	53.42	94.61
		FPR	33.19	32.87
		G-Mean	59.74	79.69

Table A.2: Hyperparameter Tuning Results for TCPC: Exact Change Point Detection Performance on the Aruba Dataset.

Context Length	Num. Future Preds	Learning Rate	TPR	FPR	G-Mean
7	2	0.0001	0.321	0.290	0.477
7	2	0.0005	0.326	0.271	0.488
7	2	0.0010	0.331	0.270	0.492
7	2	0.0010	0.293	0.277	0.461
7	3	0.0001	0.326	0.278	0.485
7	3	0.0005	0.304	0.282	0.468
7	3	0.0010	0.332	0.289	0.486
7	4	0.0001	0.287	0.257	0.462
7	4	0.0005	0.293	0.269	0.463
7	4	0.0010	0.320	0.257	0.486
7	5	0.0001	0.307	0.237	0.472
7	5	0.0005	0.313	0.245	0.480
7	5	0.0010	0.315	0.240	0.483
10	4	0.0001	0.303	0.237	0.476
10	4	0.0005	0.309	0.245	0.479
10	4	0.0010	0.319	0.237	0.488
10	5	0.0001	0.293	0.237	0.462
10	5	0.0005	0.308	0.236	0.476
10	5	0.0010	0.324	0.246	0.488
10	6	0.0001	0.295	0.241	0.466
10	6	0.0005	0.300	0.240	0.470
10	6	0.0010	0.315	0.248	0.481
13	5	0.0001	0.312	0.252	0.472
13	5	0.0005	0.297	0.248	0.473
13	5	0.0010	0.295	0.257	0.468
13	6	0.0001	0.311	0.265	0.478
13	6	0.0005	0.316	0.262	0.483
13	6	0.0010	0.304	0.264	0.473
13	7	0.0001	0.294	0.280	0.460
13	7	0.0005	0.303	0.271	0.470
13	7	0.0010	0.318	0.274	0.480

Table A.3: Hyperparameter Tuning Results for TCPC: ± 5 s Tolerance Change Point Detection Performance on the Aruba Dataset.

Context Length	Num. Future Preds	Learning Rate	TPR	FPR	G-Mean
7	2	0.0001	0.910	0.286	0.806
7	2	0.0005	0.909	0.267	0.816
7	2	0.0010	0.904	0.266	0.814
7	2	0.0010	0.907	0.272	0.812
7	3	0.0001	0.912	0.273	0.814
7	3	0.0005	0.909	0.277	0.811
7	3	0.0010	0.927	0.284	0.815
7	4	0.0001	0.904	0.252	0.822
7	4	0.0005	0.905	0.265	0.816
7	4	0.0010	0.908	0.257	0.820
7	5	0.0001	0.898	0.233	0.812
7	5	0.0005	0.903	0.235	0.816
7	5	0.0010	0.909	0.240	0.819
10	4	0.0001	0.896	0.232	0.814
10	4	0.0005	0.901	0.239	0.818
10	4	0.0010	0.902	0.247	0.816
10	5	0.0001	0.894	0.230	0.813
10	5	0.0005	0.901	0.238	0.818
10	5	0.0010	0.905	0.243	0.820
10	6	0.0001	0.897	0.239	0.813
10	6	0.0005	0.895	0.241	0.813
10	6	0.0010	0.901	0.247	0.817
13	5	0.0001	0.883	0.247	0.816
13	5	0.0005	0.882	0.244	0.817
13	5	0.0010	0.886	0.253	0.814
13	6	0.0001	0.900	0.260	0.816
13	6	0.0005	0.897	0.258	0.816
13	6	0.0010	0.902	0.260	0.817
13	7	0.0001	0.906	0.275	0.810
13	7	0.0005	0.899	0.267	0.812
13	7	0.0010	0.896	0.270	0.809

REFERENCES

- [1] Fortune Business Insights, *Smart home market size, share & industry analysis, by device type (safety & security access control, home appliances, hvac, lighting control, smart entertainment devices, smart kitchen appliances, and others), by application (retrofit and new construction), by protocol (wired and wireless), and regional forecast, 2024-2032*, 2024.
- [2] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019, The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019), The 14th International Conference on Future Networks and Communications (FNC-2019), The 9th International Conference on Sustainable Energy Information Technology.
- [3] S. G. Dhekane and T. Ploetz, "Transfer learning in human activity recognition: A survey," *arXiv preprint arXiv:2401.10185*, 2024.
- [4] M. Thukral, S. G. Dhekane, S. K. Hiremath, H. Haresamudram, and T. Ploetz, "Layout-agnostic human activity recognition in smart homes through textual descriptions of sensor triggers (tdost)," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 9, no. 1, pp. 1–38, 2025.
- [5] D. Liciotti, M. Bernardini, L. Romeo, and E. Frontoni, "A sequential deep learning application for recognising human activities in smart homes," *Neurocomputing*, vol. 396, pp. 501–513, 2020.
- [6] J. Wan, M. J. O'Grady, and G. M. P. O'Hare, "Dynamic sensor event segmentation for real-time activity recognition in a smart home context," *Personal and Ubiquitous Computing*, vol. 19, pp. 287–301, 2015.
- [7] H. Najeh, C. Lohr, and B. Leduc, "Dynamic segmentation of sensor events for real-time human activity recognition in a smart home context," *Sensors*, vol. 22, no. 14, 2022.
- [8] U. Bermejo, A. Almeida, A. Bilbao-Jayo, and G. Azkune, "Embedding-based real-time change point detection with application to activity segmentation in smart home time series data," *Expert Systems with Applications*, vol. 185, p. 115 641, 2021.
- [9] S. Aminikhanghahi, T. Wang, and D. J. Cook, "Real-time change point detection with application to smart home time series data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 1010–1023, 2019.
- [10] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems*, vol. 51, no. 2, pp. 339–367, 2017.
- [11] T. Plotz *et al.*, "Automatic detection of song changes in music mixes using stochastic models," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, 2006, pp. 665–668.
- [12] I. Cowan, G. Tesauro, J. Alspector, A. Information, and T. Plate, "Estimating analogical similarity by dot-products of holographic reduced representations.," Aug. 2000.
- [13] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
- [14] R. A. Hamad, W. L. Woo, B. Wei, and L. Yang, "Overview of human activity recognition using sensor data," in *UK Workshop on Computational Intelligence*, Springer, 2022, pp. 380–391.

- [15] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.
- [16] R. M. Schmidt, "Recurrent neural networks (rnns): A gentle introduction and overview," *arXiv preprint arXiv:1912.05911*, 2019.
- [17] R. C. Staudemeyer and E. R. Morris, "Understanding lstm—a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.
- [18] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, 2012.
- [19] C. Wang, X. Li, T. Zhou, and Z. Cai, "Unsupervised time series segmentation: A survey on recent advances," *Computers, Materials and Continua*, vol. 80, no. 2, pp. 2657–2673, 2024.
- [20] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "Segmenting time series: A survey and novel approach," *Data Mining in Time Series Databases*, vol. 57, Mar. 2003.
- [21] S. Gharghabi, Y. Ding, C.-C. M. Yeh, K. Kamgar, L. Ulanova, and E. Keogh, "Matrix profile viii: Domain agnostic online semantic segmentation at superhuman performance levels," in *2017 IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 117–126.
- [22] J. Gui *et al.*, "A survey on self-supervised learning: Algorithms, applications, and future trends," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [23] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, 2021.
- [24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PmLR, 2020, pp. 1597–1607.
- [25] J.-B. Grill *et al.*, "Bootstrap your own latent—a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.
- [26] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International conference on machine learning*, PMLR, 2021, pp. 12 310–12 320.
- [27] P. Khosla *et al.*, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.
- [28] H. Haresamudram, I. Essa, and T. Plötz, "Contrastive predictive coding for human activity recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 2, Jun. 2021.
- [29] S. Deldari, D. V. Smith, H. Xue, and F. D. Salim, "Time series change point detection with self-supervised contrastive predictive coding," in *Proceedings of the web conference 2021*, 2021, pp. 3124–3135.
- [30] A. Paszke *et al.*, *Pytorch: An imperative style, high-performance deep learning library*, 2019. arXiv: 1912.01703 [cs.LG].
- [31] C. R. Harris *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

- [32] P. Virtanen *et al.*, “Scipy 1.0: Fundamental algorithms for scientific computing in python,” *Nature Methods*, vol. 17, no. 3, pp. 261–272, Feb. 2020.
- [33] F. Pedregosa *et al.*, *Scikit-learn: Machine learning in python*, 2018. arXiv: 1201.0490 [cs.LG].
- [34] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, pp. 2673–2681, 1997.
- [35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [36] A. Mao, M. Mohri, and Y. Zhong, “Cross-entropy loss functions: Theoretical analysis and applications,” in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 23–29 Jul 2023, pp. 23 803–23 828.
- [37] D. Harbecke, Y. Chen, L. Hennig, and C. Alt, “Why only micro-f1? class weighting of measures for relation classification,” *arXiv preprint arXiv:2205.09460*, 2022.
- [38] A. Karpekov, S. Chernova, and T. Plötz, “Discover: Data-driven identification of sub-activities via clustering and visualization for enhanced activity recognition in smart homes,” *arXiv preprint arXiv:2503.01733*, 2025.