

SHAR-MAE: A Masked-Autoencoder Transformer Framework with Improved SMOTE Balancing for Imbalanced Human Activity Recognition and Real-Time Health Monitoring

Chodipalli Lakshmi Laya¹, Dr. Kondapalli Venkata Ramana²

¹PG Scholar, Department of Computer Science and Systems Engineering, Andhra University College of Engineering (A), Visakhapatnam, Andhra Pradesh, India

²Associate Professor, Department of Computer Science and Systems Engineering, Andhra University College of Engineering (A), Visakhapatnam, Andhra Pradesh, India

Abstract—Wearable inertial sensing provides a practical source of data for human activity recognition (HAR), yet supervised models are affected by the cost of annotation and by unequal activity distributions. This work introduces SHAR-MAE, a one-dimensional masked-autoencoder framework that combines an iSMOTE-based balancing procedure with Transformer representation learning. Each UCI HAR window is represented by nine synchronized sensor channels over 128 time samples. The sequence is partitioned into sixteen temporal patches, and 75% of the patches are hidden during self-supervised pretraining. Four visible tokens are processed by a four-block Transformer encoder using Layer Normalization, four-head multi-head self-attention, residual connections, and a feed-forward network. A lightweight two-block decoder reconstructs the missing patches, with mean-squared error evaluated only over the masked regions. Following pretraining, the decoder is removed and the encoder is transferred to a downstream representation module. Global average pooling and a 128-to-256 projection produce the feature vector used by a 256-to-64-to-6 classifier. On the recorded UCI HAR experiment, the system obtained 92.20% test accuracy and a macro F1-score of 0.9229 after iSMOTE balancing. The study also separates the proposed Transformer path from the original SHAR causal-convolution/Lambda-attention implementation, improving architectural clarity and reproducibility.

Index Terms—class imbalance, human activity recognition, iSMOTE, inertial sensors, masked autoencoder, multi-head self-attention, self-supervised learning, Transformer.

I. INTRODUCTION

Human Activity Recognition (HAR) using inertial sensors has become relevant to applications such as assisted living, fitness analysis, remote monitoring, and activity-aware healthcare systems. Smartphone and wearable sensors continuously capture acceleration and angular-motion signals, making it possible to infer activities from short temporal windows. A major practical difficulty is that obtaining reliable activity labels for every sensor window is expensive, while naturally collected datasets may contain unequal numbers of examples for different activities.

Self-supervised learning offers a way to learn useful representations before task-specific labels are applied. Instead of predicting an externally supplied label during pretraining, the model is given a reconstruction or prediction task derived from the signal itself. In the proposed SHAR-MAE pipeline, temporal portions of a sensor window are hidden and the network learns to recover the missing information from the visible context.

The work changes two central elements of the reference SHAR design. First, the contrastive pretext objective is replaced with masked reconstruction. Second, the original causal convolution/Lambda-attention feature extractor is replaced by a Transformer-based MAE encoder. The original SHAR encoder is retained as a reference

implementation but is not used in the MAE forward path that produced the reported experiment.

The principal contributions are: (1) an iSMOTE-assisted balancing stage for the multichannel training windows; (2) a one-dimensional masked-autoencoder architecture for inertial sequences; (3) an explicit four-block Transformer encoder with LayerNorm, self-attention, residual paths, and feed-forward layers; (4) a clear transfer procedure from self-supervised pretraining to supervised classification; and (5) an application layer that maps predicted activities to health-tracking indicators..

II. RELATED WORK

A. Deep Learning for Human Activity Recognition

Convolutional networks are widely used for sensor HAR because convolution kernels can learn local temporal motifs. Recurrent architectures such as LSTM networks model sequential dependencies, while attention-based models provide a mechanism for relating information at different temporal positions. These approaches can perform well, but conventional supervised training generally depends on labelled activity examples.

B. Self-Supervised Representation Learning

Contrastive methods learn by bringing related representations together and separating unrelated ones, often using augmented views of the same observation. Masked reconstruction follows a different principle: part of the input is intentionally removed and the model is trained to predict the missing content. For sensor signals, this creates a reconstruction target directly from the recorded sequence and does not require an activity label for the reconstruction loss.

C. Self-Supervised HAR

Earlier HAR research has investigated transformation-based objectives, contrastive predictive coding, and masked reconstruction. SHAR-MAE applies the masked-autoencoder idea to short multichannel inertial windows and couples the representation-learning stage with an imbalance-handling procedure. The resulting design is intended to study whether a Transformer can replace the reference SHAR representation backbone.

D. Imbalanced Learning and iSMOTE

Standard SMOTE creates additional minority observations by interpolation between samples from the same class. The implemented iSMOTE routine adds a neighbour-consistency check after a candidate is generated. A synthetic candidate is retained only when its five nearest neighbours in the complete training set all belong to the intended class. This additional check is intended to reject candidates located near class-overlap regions.

III. PROPOSED SHAR-MAE METHODOLOGY

The proposed workflow has two stages. In the first stage, the training windows are balanced and used for self-supervised masked reconstruction. In the second stage, the pretrained encoder is transferred to a classification network and fine-tuned using activity labels. For the UCI HAR experiment, every input example is a 9×128 tensor: nine synchronized channels and 128 temporal samples.

Figure 1. Proposed SHAR-MAE architecture for multichannel inertial HAR

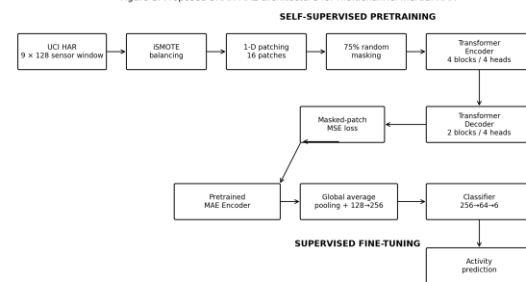


Fig. 1. End-to-end SHAR-MAE pipeline from sensor windows to downstream activity prediction.

A. Sensor Window and Input Representation

The input contains body acceleration on the x, y and z axes, body angular velocity on the x, y and z axes, and total acceleration on the x, y and z axes. The implementation preserves these measurements as a 9×128 tensor. The UCI HAR training partition contains 7,352 windows and the held-out test partition contains 2,947 windows. At 50 Hz, a 128-sample window spans 2.56 seconds.

B. iSMOTE Balancing

For a minority class c , a synthetic sample is generated by interpolating between two same-class observations. The implemented operation can be expressed as

$$x_{new} = x_i + \lambda(x_j - x_i), \quad \lambda \sim U(0,1).$$

The generated point is then tested using K-nearest-neighbour validation over the complete training population. K is set to 5. The point is accepted only when all five neighbours carry the target label. In the recorded run, the balanced dataset contains 1,407 windows per class, giving 8,442 training windows in total.

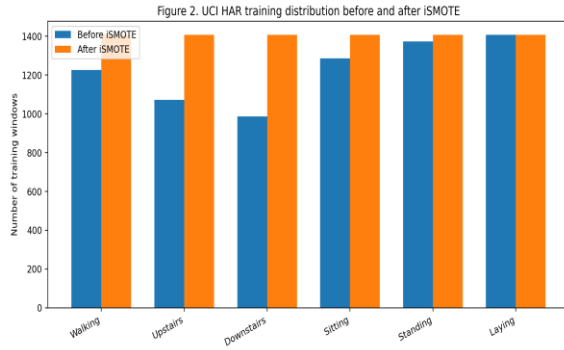


Fig. 2. UCI HAR training distribution before and after the implemented iSMOTE balancing step.

The recorded procedure added 181, 334, 421, 121 and 33 examples to Walking, Walking Upstairs, Walking Downstairs, Sitting and Standing, respectively. Laying already contained the largest number of training windows and therefore required no synthetic addition.

C. One-Dimensional Patch Embedding

Rather than presenting every sensor sample as an independent token, the model groups consecutive time samples into short temporal patches. A patch size of eight divides the 128-sample sequence into sixteen non-overlapping patches. Because all nine channels are retained within each patch, a patch contains $9 \times 8 = 72$ scalar values. A learned linear projection converts each 72-dimensional patch into a 128-dimensional Transformer token.

$$p_i \in \mathbb{R}^{72} \rightarrow e_i = W_p p_i + b_p \in \mathbb{R}^{128}$$

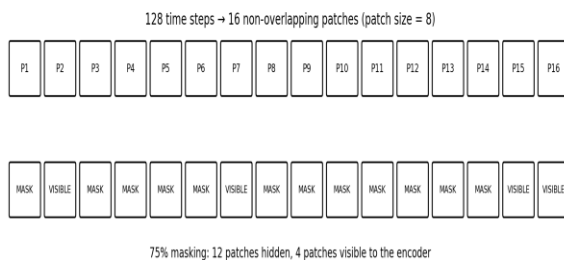


Fig. 3. Temporal patch formation and 75% random patch masking.

D. Positional Encoding

Self-attention does not by itself specify the original order of tokens. The implementation therefore adds fixed sinusoidal positional information to the sixteen patch embeddings before masking. For position p and embedding dimension d , the encoding follows

$$PE(p, 2k) = \sin(p/10000^{(2k/d)}),$$

$$PE(p, 2k+1) = \cos(p/10000^{(2k/d)}).$$

The fixed positional representation lets the encoder distinguish patches according to their original temporal location.

E. Random Patch Masking

The MAE pretext task hides 75% of the sixteen patch tokens. Consequently, four tokens remain visible while twelve are withheld from the encoder. A separate random permutation is generated for each sample, and restoration indices are retained for the decoder. Importantly, the encoder never processes the twelve hidden tokens during pretraining. This is different from the older time-step masking utility in the codebase, which uses a much smaller default masking probability and masks individual time positions rather than complete temporal patches.

F. Transformer Encoder Block

The encoder contains four identical pre-norm Transformer blocks operating on 128-dimensional tokens with four attention heads. Each block follows a repeated sequence: LayerNorm, multi-head self-attention, residual addition, a second LayerNorm, a feed-forward network, and a second residual addition.

Figure 4. Pre-norm Transformer block used in the SHAR-MAE encoder and decoder

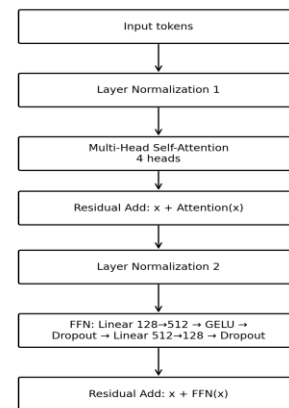


Fig. 4. Internal sequence of operations in the pre-norm Transformer block.

The first LayerNorm normalizes the token representation before attention. Multi-head self-attention then allows each visible token to use information from the other visible tokens. The first residual path adds the attention result to the original block input. A second LayerNorm precedes the feed-forward sublayer, which expands the embedding from 128 to 512 and then projects it back to 128 using GELU activation and dropout of 0.1. The second residual path adds the feed-forward output to its input.

G. Multi-Head Self-Attention

Figure 5. Multi-head self-attention operation

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k})V$$



Each Transformer block uses 4 attention heads; attention weights are learned from the visible patch-token sequence.

Fig. 5. Multi-head self-attention used in the Transformer encoder.

For a token matrix X , learned projections generate queries, keys, and values. The scaled dot-product attention used in each head is

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k})V$$

Four heads are used so that the model can learn several interaction patterns in parallel. Since Q , K , and V originate from the same sequence, the operation is self-attention. For HAR, this enables a temporal segment to incorporate information from other visible segments when forming its representation.

H. MAE Decoder and Reconstruction Loss

Only masked patches contribute to the reconstruction loss

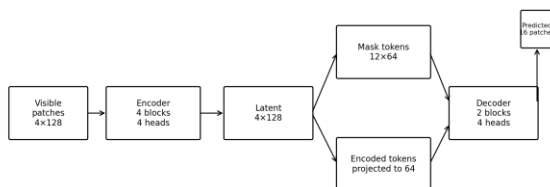


Fig. 6. Asymmetric MAE encoder-decoder used during self-supervised pretraining.

The four visible encoder tokens are projected from 128 to 64 dimensions. Twelve learnable mask tokens are inserted at the missing positions and the full sixteen-token order is restored. Two decoder Transformer blocks process this sequence. The decoder output is mapped to 72 values per patch, matching the original 9×8 patch size.

$$L_{MAE} = (1/|M|) \sum_{i \in M} ||\hat{x}_i - x_i||^2$$

Here M denotes the set of hidden patch positions, x_i is the original patch, and $\hat{x}_i$ is its reconstruction. Computing the loss only over hidden patches makes the pretext task focus on recovering information that was not supplied to the encoder.

I. MAE Architecture Specification

Table I. MAE architecture specification

Component	Implementation	Role / output
Input	9 channels $\times$ 128 samples	Sensor window
Patch embedding	Patch size 8; 72 $\rightarrow$ 128	16 tokens $\times$ 128
Position encoding	Fixed sinusoidal	Temporal order
Masking	Random; 75%	4 visible + 12 hidden
Encoder	4 Transformer blocks; 4 heads	Latent representation
Encoder block	LN $\rightarrow$ MSA $\rightarrow$ residual $\rightarrow$ LN $\rightarrow$ FFN $\rightarrow$ residual	128-D token refinement
FFN	128 $\rightarrow$ 512 $\rightarrow$ 128; GELU; dropout 0.1	Nonlinear transformation
Decoder input	128 $\rightarrow$ 64 plus 12 mask tokens	16 tokens $\times$ 64
Decoder	2 Transformer blocks; 4 heads	Reconstruction
Prediction layer	64 $\rightarrow$ 72	Reconstructed patch
Objective	Masked-patch MSE	Self-supervised loss

J. Encoder Transfer for Fine-Tuning

Once pretraining is complete, the decoder is discarded. A separate MAEEncoder keeps the patch embedding, sinusoidal position representation, four Transformer blocks, and final LayerNorm. The

recorded transfer routine copies the corresponding pretrained parameters into this downstream encoder. The additional 128-to-256 projection is newly initialized because it is not part of the MAE pretraining encoder.

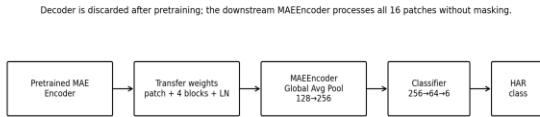


Fig. 7. Transfer of the pretrained MAE encoder into the downstream classification path.

During fine-tuning and inference, masking is disabled and all sixteen patches are processed. Global average pooling combines the sixteen token representations into a 128-dimensional vector. A linear 128-to-256 layer followed by ReLU produces the 256-dimensional representation required by the classifier.

K. Classification Head

256 → Linear → 64 → ReLU → Dropout(0.2) → Linear → 6

The six logits represent Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, and Laying. The class associated with the largest logit is selected as the predicted activity.

L. Separation from the Original SHAR Backbone

The source implementation also contains the reference SHAR encoder, which uses causal one-dimensional convolutions followed by a Lambda-based attention layer and a fully connected projection. That encoder is not part of the reported MAE-SHAR forward path. The proposed path is instead PatchEmbed1D, four Transformer blocks, final LayerNorm, global average pooling, and a 128-to-256 projection. Keeping these two paths separate is important when the experiment is reproduced or when a future controlled comparison is performed.

IV. ARCHITECTURAL DIFFERENCES BETWEEN BASE SHAR AND MAE-SHAR

The two systems address the same six-class HAR task and use the same iSMOTE concept, but they employ different representation-learning mechanisms. The following comparison describes the

implemented distinction without treating the current experiment as proof that one backbone universally outperforms the other.

Table II. Architectural differences between base SHAR and MAE-SHAR

Component	Base SHAR	Proposed MAE-SHAR
Input	9×128 sensor window	9×128 sensor window
Balancing	iSMOTE	iSMOTE
Signal corruption	Random time-step masking	Random patch masking (75%)
Temporal representation	Causal 1-D CNN	1-D patch embedding
Long-range modeling	Lambda attention	Multi-head self-attention
Pretext objective	Contrastive / NT-Xent-style	Masked reconstruction / MSE
Pretraining backbone	Causal CNN + Lambda	4-block Transformer encoder
Decoder	No dedicated MAE decoder	2-block Transformer decoder
Transferred encoder	CNN/Lambda encoder	MAE Encoder Transformer
Downstream representation	256-D	128-D pooled → 256-D
Classifier	256→64→6	256→64→6

A. Why the Proposed Path Does Not Include the Original CNN

The exclusion of the causal CNN is a methodological choice. The experiment is intended to examine a Transformer-based masked-autoencoder representation learner as a replacement for the original SHAR feature extractor. Convolution is effective for local temporal motifs, but adding the original convolutional front end would make it difficult to isolate the effect of the MAE-Transformer representation. CNNs therefore remain relevant as reference or hybrid baselines, but they are not part of the proposed forward path used for the reported result.

B. Why Transformer Self-Attention Is Used

Human activities can contain motion relationships separated in time. A walking segment, for example,

may contain repeated acceleration and rotational patterns whose interpretation benefits from context elsewhere in the same window. Self-attention permits direct interaction among the available temporal tokens, while the MAE objective supplies a natural reconstruction task after tokens are hidden.

The four-head design allows several learned interaction subspaces to be considered simultaneously. LayerNorm supports stable optimization, residual paths preserve information and facilitate gradient propagation, and the feed-forward network performs token-wise nonlinear transformation. These components together constitute the complete Transformer block rather than attention alone.

C. Meaning of the Nine Sensor Channels

The nine channels are synchronized inertial measurements rather than unrelated tabular attributes. Three channels describe body acceleration, three describe angular velocity, and three describe total acceleration. Their combined tensor has dimensions 9×128 . Retaining all three groups allows the model to use complementary information about translation, rotation, and overall acceleration.

For iSMOTE, each 9×128 window is flattened to 1,152 values only for nearest-neighbour calculations and synthetic generation. The neural network receives the restored 9×128 structure. Each temporal patch contains all nine channels over eight samples, so cross-channel information remains inside every Transformer token.

D. Why Temporal Grouping Is Necessary

The order of sensor observations carries information about movement dynamics. Treating the nine channels as independent tabular columns would discard the temporal relationships that distinguish one motion pattern from another. Patch embedding therefore groups neighbouring time samples, positional encoding preserves their order, and self-attention models relationships among the resulting temporal units.

E. End-to-End Data Flow

The implemented sequence is: 9-channel sensor window $\rightarrow$ iSMOTE balancing $\rightarrow$ sixteen temporal patches $\rightarrow$ 75% random patch masking $\rightarrow$ four visible tokens $\rightarrow$ four-block Transformer encoder $\rightarrow$

decoder with mask tokens $\rightarrow$ masked-patch reconstruction $\rightarrow$ MSE loss. After pretraining, the decoder is removed, all sixteen patches are processed by the transferred encoder, global average pooling generates the representation, the 128-to-256 projection expands the feature vector, and the 256-to-64-to-6 classifier produces the activity prediction.

F. Research Question

The study asks whether a Transformer-based masked autoencoder, when combined with iSMOTE-based balancing, can replace the reference SHAR causal-convolution/Lambda representation backbone while learning useful representations for imbalanced multichannel inertial HAR. This question can be tested more rigorously through controlled comparisons of pretraining, masking, backbone, and label-availability settings.

V. TRAINING PROCEDURE

A. Self-Supervised Pretraining

The UCI HAR training partition is first processed by iSMOTE with $K=5$. The activity labels are used by the balancing routine to construct the training distribution, but the MAE reconstruction target is obtained directly from the sensor windows. The balanced windows are then presented to MAE1D for 80 epochs with batch size 128. The recorded implementation uses AdamW, learning rate 1.5×10^{-3} , weight decay 0.05, cosine annealing, a 75% masking ratio, and dropout 0.1.

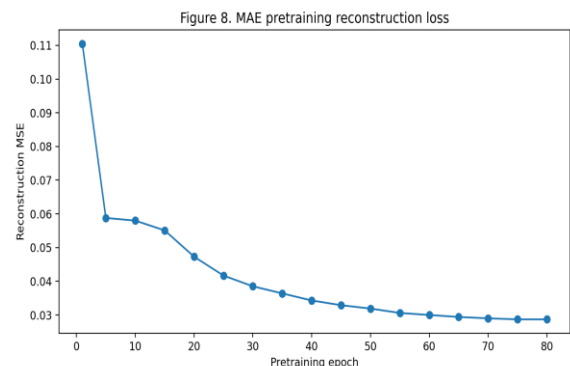


Fig. 8. Recorded MAE pretraining reconstruction-loss trajectory.

The logged reconstruction MSE decreased from 0.1105 at the first epoch to 0.0287 at epoch 80. The

complete MAE checkpoint and the extracted encoder checkpoint were saved after pretraining.

B. Supervised Fine-Tuning

The implementation supports a label-ratio parameter. When less than one, a stratified subset is selected with `random_state=42` and is subsequently rebalanced using `iSMOTE`. For the reported experiment, the ratio is 1.0, so all 7,352 labelled training windows are used and balancing produces 8,442 windows.

The pretrained MAEEncoder is connected to SHAR_Classifier and optimized end-to-end for 60 epochs with batch size 128. The recorded configuration uses AdamW with learning rate 5×10^{-4} , weight decay 0.01, cosine annealing, and cross-entropy with label smoothing $\epsilon=0.05$. The current implementation retains the checkpoint associated with the highest test accuracy during training; this is a methodological limitation because a separate validation set should be used for checkpoint selection in a publication-quality evaluation.

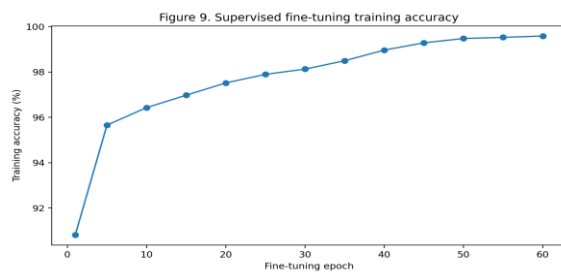


Fig. 9. Recorded supervised fine-tuning training-accuracy trajectory.

C. Mathematical Form of Fine-Tuning

Let $f\theta(x)$ denote the transferred MAEEncoder and $g\phi$ denote the classifier. The network output can be written as $z = g\phi(f\theta(x))$, where z contains six activity logits. The supervised objective updates both the transferred encoder parameters and the newly initialized projection/classification parameters.

VI. EXPERIMENTAL SETUP

A. UCI HAR Dataset

The UCI HAR dataset contains recordings from 30 subjects performing six activities with a waist-mounted smartphone. The implementation uses nine inertial channels sampled at 50 Hz and fixed

windows of 128 samples. The original subject-disjoint training and test partitions are retained.

Table III. UCI HAR class distribution in the training and test partitions

Activity	Train	Test
Walking	1,226	496
Walking Upstairs	1,073	471
Walking Downstairs	986	420
Sitting	1,286	491
Standing	1,374	532
Laying	1,407	537
Total	7,352	2,947

B. Hyperparameters

Table IV. Hyperparameters used for MAE pretraining and fine-tuning

Parameter	MAE pretraining	Fine-tuning
Batch size	128	128
Epochs	80	60
Learning rate	1.5×10^{-3}	5×10^{-4}
Optimizer	AdamW	AdamW
Weight decay	0.05	0.01
Scheduler	CosineAnnealingLR	CosineAnnealingLR
Patch size	8	8
Embedding dimension	128	128
Encoder depth	4 blocks	4 blocks
Attention heads	4	4
Mask ratio	75%	None
Decoder dimension	64	Not used
Decoder depth	2 blocks	Not used
Dropout	0.1	0.1 encoder; 0.2 classifier
Label smoothing	Not used	0.05
iSMOTE K	5	5

C. Evaluation Measures

The recorded evaluation uses overall accuracy, macro-averaged F1, and class-wise precision, recall, and F1. Macro F1 is useful for an imbalanced recognition task because each activity contributes equally to the average rather than allowing the largest class to dominate the summary measure.

VII. RESULTS AND DISCUSSION

A. Effect of iSMOTE

The balancing stage increased the training set from 7,352 to 8,442 windows. The final distribution contained 1,407 examples for each activity. The added observations were 181 for Walking, 334 for Walking Upstairs, 421 for Walking Downstairs, 121 for Sitting, and 33 for Standing; Laying did not require additional samples.

B. Classification Performance

The recorded 60-epoch fine-tuning run produced 92.20% test accuracy and a macro F1-score of 0.9229. The reported test loss was 0.5211.

Table V. Class-wise classification report on the UCI HAR test set

Class	Precision	Recall	F1	Support
Walking	0.99	0.96	0.98	496
Walking Upstairs	0.97	0.94	0.96	471
Walking Downstairs	0.95	0.98	0.96	420
Sitting	0.82	0.78	0.80	491
Standing	0.82	0.87	0.85	532
Laying	0.99	1.00	1.00	537
Macro average	0.92	0.92	0.92	2,947

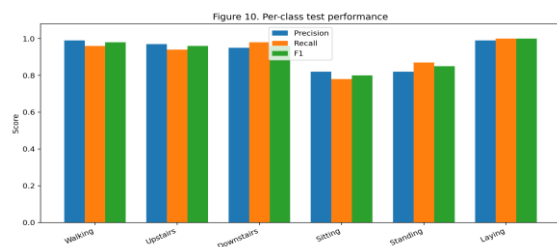


Fig. 10. Recorded class-wise precision, recall and F1-score on the UCI HAR test set.

The locomotion activities and Laying show higher F1 values in the recorded evaluation, whereas Sitting and Standing are comparatively harder to separate. This pattern is plausible for inertial HAR because low-motion postures can produce similar signal characteristics. The class-wise results also show why macro F1 should accompany overall accuracy.

C. Interpretation of the Learned Representation

The reconstruction-loss curve indicates that the model learned to recover hidden temporal patches from visible context during pretraining. Because a visible token can attend to other visible tokens, information from separated parts of the window can contribute to reconstruction. In the downstream stage, the complete sixteen-patch sequence is encoded and pooled into a 256-dimensional representation.

Training accuracy increased from 90.81% at epoch 1 to 99.59% at epoch 60 on the balanced training set, while test accuracy was 92.20%. The difference indicates that additional repeated runs, validation-based checkpointing, and ablation studies are needed before drawing a stronger architectural conclusion.

VIII. ABLATION AND EXTENDED EXPERIMENT PLAN

The current run demonstrates feasibility but does not establish that the Transformer MAE is universally better than the reference SHAR backbone. The following experiments are therefore proposed as extensions rather than results already obtained.

Table VI. Proposed ablation and extended experiments

Study	Variants	Purpose
Balancing	Without iSMOTE vs iSMOTE	Measure balancing contribution
Pretraining	From scratch vs MAE pretraining	Measure representation benefit
Mask ratio	25%, 50%, 75%, 90%	Assess masking sensitivity
Encoder depth	2, 4, 6 blocks	Assess model capacity
Attention heads	2, 4, 8	Assess head sensitivity
Patch size	4, 8, 16	Assess temporal granularity
Label availability	10%, 25%, 50%, 75%, 100%	Study label efficiency
Pretext task	Contrastive baseline vs MAE	Compare self-supervised objectives
Backbone	CNN/LSTM baseline vs Transformer MAE	Compare representation choices

A label-ratio study is especially relevant to the motivation for self-supervised learning. The test set should remain fixed while different fractions of labelled training data are used, and repeated random seeds should ideally be reported with mean and standard deviation.

IX. EXPLAINABILITY AND HEALTH-TRACKING APPLICATION

The current implementation includes a downstream dashboard driven by predicted activity labels. Its application modules include calorie estimation, step counting, sedentary/active-time budgeting, sleep-stage estimation, fall detection, and heart-rate/HRV-based stress estimation. These application outputs should be distinguished from the HAR benchmark because some modules use simulated or rule-based signals rather than clinically measured physiological data.

For stronger explainability, future experiments can visualize sensor-channel and temporal-region contributions using attention visualization, input perturbation, integrated gradients, or SHAP-style attribution. Such analyses should be treated as explanatory evidence and should not be presented as direct clinical measurements without appropriate validation.

X. LIMITATIONS AND THREATS TO VALIDITY

Several limitations should be considered. First, the principal experiment uses the complete labelled training set, so the claimed advantage of self-supervision under scarce labels has not yet been demonstrated. Second, only UCI HAR is quantitatively evaluated in the current version. Third, the training script selects the best checkpoint using test accuracy; a validation partition should replace this procedure for a rigorous publication protocol. Fourth, iSMOTE is applied before MAE pretraining using training labels to construct the balanced distribution. Although the reconstruction loss itself does not use activity labels, the data-selection process is label-informed and should therefore be described accurately. Finally, the health dashboard is not clinically validated in the current study, and physiological outputs such as HRV- or sleep-related

indicators require comparison against appropriate measured references before clinical interpretation.

XI. CONCLUSION

This paper presented SHAR-MAE, a one-dimensional masked-autoencoder framework for imbalanced human activity recognition. The system combines iSMOTE balancing with Transformer-based self-supervised representation learning. Each 9×128 sensor window is converted into sixteen 128-dimensional temporal tokens, twelve are masked during pretraining, and the four visible tokens are processed by four pre-norm Transformer blocks. A two-block decoder reconstructs the hidden patches using a masked MSE objective.

After pretraining, the decoder is removed and the encoder is transferred to a downstream classification module. All patches are then processed, pooled, projected from 128 to 256 dimensions, and passed through a 256-to-64-to-6 classifier. On the recorded UCI HAR experiment, the system achieved 92.20% test accuracy and 0.9229 macro F1. The results support the feasibility of the implemented architecture, while the documented limitations indicate that repeated runs, validation-based model selection, label-ratio experiments, ablations, and cross-dataset evaluation are needed for stronger publication claims.

REFERENCES

- [1] R. A. Hamad, W. L. Woo, B. Wei, and L. Yang, "Self-supervised learning for activity recognition based on datasets with imbalanced classes," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 9, no. 5, pp. 3441–3454, Oct. 2025.
- [2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and T. W. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
- [3] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, art. no. 115, 2016.
- [4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive

- learning of visual representations,” in Proc. ICML, 2020, pp. 1597–1607.
- [5] A. Saeed, T. Ozcelebi, and J. Lukkien, “Multi-task self-supervised learning for human activity detection,” Proc. ACM IMWUT, vol. 3, no. 2, 2019.
- [6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
- [7] H. Haresamudram et al., “Masked reconstruction based self-supervision for human activity recognition,” in Proc. ISWC, 2020, pp. 45–49.
- [8] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in Proc. IEEE/CVF CVPR, 2022, pp. 16000–16009.
- [9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
- [10] R. A. Hamad, M. Kimura, L. Yang, W. L. Woo, and B. Wei, “Dilated causal convolution with multi-head self attention for sensor human activity recognition,” Neural Comput. Appl., vol. 33, pp. 13705–13722, 2021.
- [11] A. Vaswani et al., “Attention is all you need,” in Proc. NeurIPS, 2017, pp. 5998–6008.
- [12] I. Bello, “LambdaNetworks: Modeling long-range interactions without attention,” in Proc. ICLR, 2021.
- [13] H. Haresamudram, I. Essa, and T. Plötz, “Contrastive predictive coding for human activity recognition,” Proc. ACM IMWUT, vol. 5, no. 2, 2021.
- [14] I. Chi et al., “SelfHAR: Improving human activity recognition through self-training with unlabeled data,” Proc. ACM IMWUT, vol. 5, no. 1, 2021.
- [15] H. Xu, P. Zhou, R. Tan, M. Li, and G. Shen, “LIMU-BERT: Unleashing the potential of unlabeled data for IMU sensing applications,” in Proc. ACM SenSys, 2021, pp. 220–233.
- [16] D. Anguita, A. Ghio, A. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in Proc. ESANN, 2013.
- [17] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in Proc. ICLR, 2019.
- [18] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep learning training by reducing internal covariate shift,” in Proc. ICML, 2015.
- [19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.