HUMAN ACTIVITY RECOGNITION WITH CONVOLUTIONAL AND MULTI-HEAD ATTENTION LAYER BASED NEURAL NETWORK

DENİZ ADALI ATLIHAN

JANUARY 2022

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER'S THESIS IN

COMPUTER ENGINEERING

HUMAN ACTIVITY RECOGNITION WITH CONVOLUTIONAL AND MULTI-HEAD ATTENTION LAYER BASED NEURAL NETWORKS

DENİZ ADALI ATLIHAN

JANUARY 2022

ABSTRACT

HUMAN ACTIVITY RECOGNITION WITH CONVOLUTIONAL AND MULTI-HEAD ATTENTION LAYER BASED NEURAL NETWORKS

ATLIHAN, Deniz Adalı

Master of Science in Computer Engineering

Supervisor: Prof. Dr. Hasan OĞUL

January 2022, 69 pages

Human Activity Recognition (HAR) refers to classifying human activities with time-series data generated by sensors. Although there are many different sensing techniques for HAR, this thesis uses wrist-worn accelerometer data provided by the HANDY dataset due to the recent development of mobile wearable sensing devices. In the proposed model, the feature extraction layer is connected to the attention layer, respectively, and this context is connected to the fully connected layer to classify the inputs. Due to its achievements in feature extraction, Convolutional Neural Network (CNN) was used in the feature extraction layer, Multi-Head Attention Layer was used after CNN to evaluate every dimension of the 3D time-series data coming from the acceleration sensor. After the feature extraction and attention layer, this model, which ended with a fully connected layer with the SoftMax classifier, reached 0. 935 validation accuracy when evaluated with categorical cross-entropy loss.

Keywords: Human Activity Recognition, Convolutional Neural Networks, Multi-head attention

ÖZ

EVRİŞİMLİ VE ÇOK KAFALI DİKKAT KATMANLI SİNİR AĞLARIYLA İNSAN AKTİVİTELERİNİ TANIMA

ATLIHAN, Deniz Adalı
Bilgisayar Mühendisliği Yüksek Lisans
Danışman: Prof. Dr. Hasan OĞUL
Ocak 2022, 69 sayfa

İnsan Aktivitesi Tanıma (HAR), sensörler tarafından oluşturulan zaman serisi verileriyle insan aktivitelerinin sınıflandırılmasını ifade eder. HAR için birçok farklı algılama tekniği olmasına rağmen, bu tezde, mobil giyilebilir algılama cihazlarındaki son gelişmeler nedeniyle, HANDY veri seti tarafından sağlanan bileğe takılan ivmeölçer verileri kullanılmıştır. Önerilen modelde, öznitelik çıkarma katmanı sırasıyla dikkat katmanına bağlanmıştır ve bu bağlam girdileri sınıflandırmak için tam bağlantılı katmana bağlanmıştır. Öznitelik çıkarımdaki başarılarından dolayı öznitelik çıkarma katmanında Evrişimsel Sinir Ağı (CNN), ivme sensöründen gelen 3B zaman serisi verilerinin her boyutunu değerlendirmek için CNN'den sonra Çok Kafalı Dikkat Katmanı kullanılmıştır. Öznitelik çıkarma ve dikkat katmanından sonra SoftMax sınıflandırıcı ile tam bağlantılı katman ile sonlanan bu model, kategorik çapraz entropi kaybı ile değerlendirildiğinde 0,935 doğrulama oranına ulaşmıştır.

Anahtar Kelimeler: İnsan Aktivitesi Tanıma, Evrişimli Sinir Ağları, Çok-kafalı dikkat.

ACKNOWLEDGEMENT

I would like to thank the founder of the Turkish Republic, Mustafa Kemal ATATÜRK, the architect of our freedom, the great leader I inspired.

I would also like to thank my mother, Canan ATLIHAN, from whom I learned what goodness and kindness, to my father Recai ATLIHAN, who protected our family in the face of all difficult circumstances, to my brother Baran ATLIHAN who made me feel that I am not alone in life, to my precious wife Sultan KILIÇ ATLIHAN who not only shared her life with me but always supported me throughout my challenging education and business life, to Şafak Burak ÇEVİKBAŞ who inspired me to develop my realistic perspective and ability to think abstractly and to my advisor Hasan OĞUL for guiding me in my thesis journey and lead to expanding my academic horizon

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOLS

Hz	: Hertz
m/s^2	: Meter per second square
$°/s^2$	: Degree per second square
mV	: Millivolt

ABBREVIATIONS

AI	: Artificial Intelligence
ANN	: Artificial Neural Network
BN	: Batch Normalization
CNN	: Convolutional Neural Network
DanHAR	: Dual Attention Network
FC	: Fully Connected
FFT	: Fast Fourier Transform
GRU	: Gated Recurrent Unit
HAR	: Human Activity Recognition
IoT	: Internet of Things
IMU	: Inertial Measurement Units
LDA	: Latent Dirichlet Allocation
LSTM	: Long Short Term Memory
ML	: Machine Learning
ReLU	: Rectified Linear Unit
RNN	: Recurrent Neural Network
Val	: Validation
Acc	: Accuracy

CHAPTER I

INTRODUCTION

1.1 PROBLEM & MOTIVATION

Human Activity Recognition (HAR) is a crucial term for sensing human activities with sensors and meaning them for different purposes such as health services, athletic performance tracking and human survey.

There are different ways of sensing techniques such as prominent vision-based [1], medical domain based [2] and radar-based [3] methods. However, in addition to recent development [4], the Internet of Things (IoT) and smart wearable devices like smartwatches have widespread usage and ease of using small wearable ones. For instance, this thesis focused on the measurement of the wrist-worn sensor.

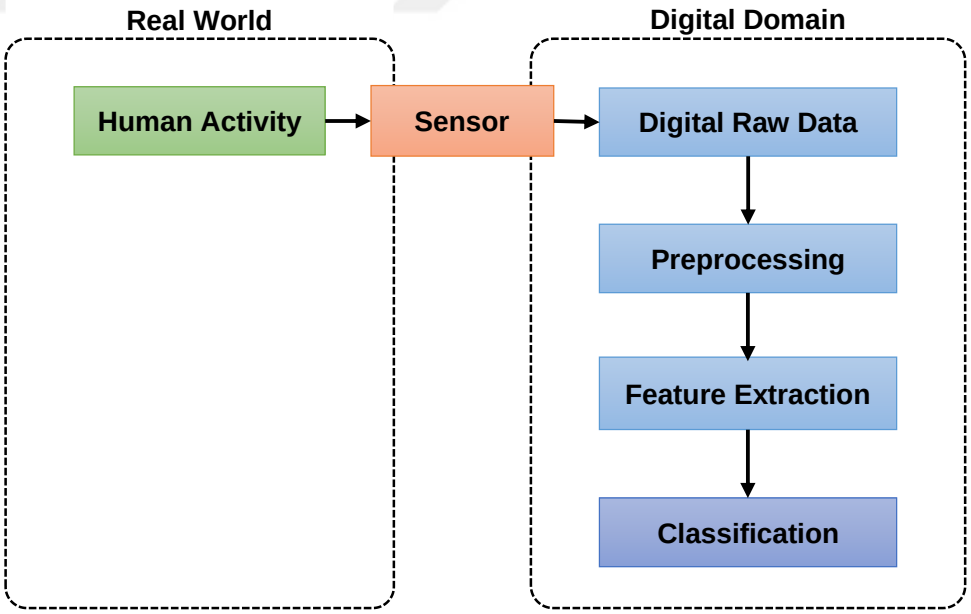


Figure 1.1: Simple HAR Application Process

A simple HAR application process can be represented as illustrated in Figure 1.1. The activity of a human can be sensed with a sensor and translated into digital data. The translated raw data is needed to be preprocessed for determining features. Due to collected sensor data being a time series, at least, this data must be segmented to time frames for classifying them. Generally, measured data is also normalized to 0 – 1 range here. In addition, in some works which used Inertial Measurement Units (IMU) sensor data, the noise filtering process is also applied in this phase due to the noisy nature of IMU sensors

In many works [5] [6] [7] [8] [9] [10] [11] [12] handcrafted features generally generated by time and frequency domain digital signal processing techniques, in contrast, in some other works, convolution process is taken precedence over signal processing [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28]. In this thesis also, the convolutional process is used for feature extraction.

In the last part, classification can be performed with machine learning or deep learning techniques. There are successful works [6] [29] [8] [9] [10] [11] which are used machine learning techniques, fully connected layers ended with SoftMax classifier also performed satisfying results.

With recent technical developments of recurrent neural networks, the performance of classifying time series improved as like as collected data via sensors during a time. Some of recent works [13] [14] [15] [16] [17] [30] [31] [18] [19] [20] [32] [25] [26] [33] used attention mechanism firstly explained with Transformer by Vaswani et a [34] l, for boosting time series classification. In this thesis, an attention mechanism is also used for boosting classification performance.

1.2 RECENT WORKS

Without a convolutional process, handcrafted feature extractions combined with machine learning techniques are used to classify human activities with sensor measurements.

In 2011, Chernbumroong et al. used time and frequency domain features of wrist-worn accelerometer data for classifying activities [29]. The work that they built Artificial Neural Network (ANN) and Decision Tree C4.5 shows that Decision Tree has better performance than ANN.

In 2012, Scholl and van Laerhoven used wrist-worn accelerometer data for determining smoking habits with signal processing methods [6]. They applied a low pass filter to derive signal data then derived attributes of the signal as mean and variance. In the end, they have applied Gaussian Classifier to the features and reached 51.2% success rate. Likewise, in 2016, Nguyen et al. used a wrist-worn accelerometer signal for classifying daily activities with time and frequency domain features [7]. Their work which used different machine learning techniques and combined also focused on sensor data generated from sensors positioned on both left and right wrists. They reached 91.2% success rate for the Random Forest method and offered the combination of the Random Forest and k-Nearest Neighbors method to improve performance. They also worked with Multi-Layer Perceptron but could not achieved a better score than Random Forest Method. Same year, Konak et al. classified daily activities with accelerometer data collected by mobile phone using time and frequency domain features [11]. In addition, they offered the Random Forest method for best accuracy and proposed using only accelerometer data over accelerometer and gyroscope data for battery usage advantage in real-life applications.

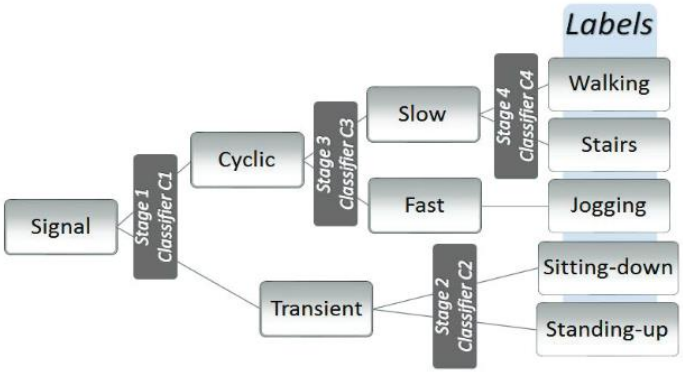
In 2018 Mehrang et al. used wristband accelerometer data and heart rate measurement together [8]. Their Random Forest based classification method reached 89.4% success rate with using handcrafted signal features.

Same year, Hegde et al. tried to classify daily living activities with combination of wrist-worn accelerometer and SmartStep sensor together [10]. Their measurement was processed with Max Relevance, Min Redundancy Algorithm [28] and classified with Multinomial Logistic Discrimination Method and reached to 94% success rate.

In 2019, Konsig et al. proposed S-PAR [35] which uses both accelerometer and angular velocity data derived from wrist-worn device. Their model focused on determining energetic activities with Support Vector Machine and dormant activities with Linear Discriminant Analysis. Both techniques have shown better performance than Random Forest for their works and S-PAR reached %88.62 ± 11.71 F-score.

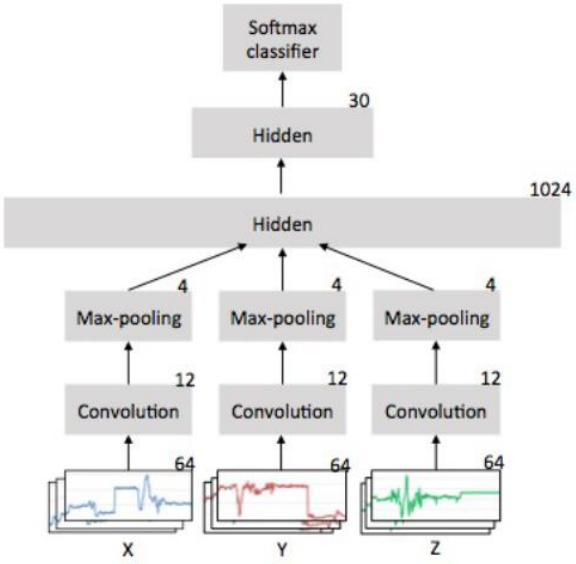
Even though recent progress in Deep Learning, Machine Learning methods are still thriving for HAR applications. In 2021 Arani et al. combined wrist-worn accelerometer and electrocardiogram measurements and generated handcrafted features [9]. Their Random Forest based approach reached 94% for subject dependent and 86.1% for subject independent F1-scores.

In 2019 Abbas et al. applied four stages approach for HAR, as shown in Figure 1.2. The handcrafted signal features used in the ANN classifier reached 98.41% accuracy [5].



Source: [5]

Figure 1.2: 4 Stages Approach of Abbas et al.

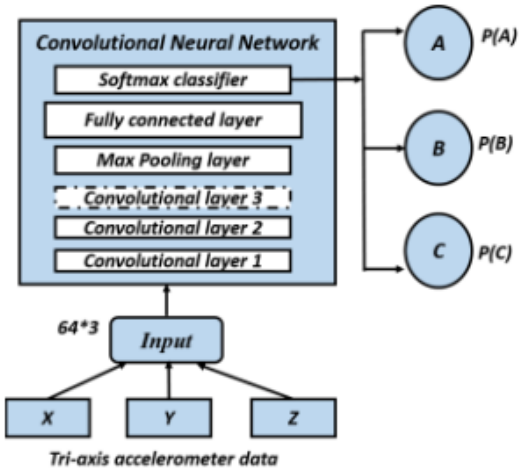


Source: [28]

Figure 1.3: CNN Architecture Offered by Zeng et al.

Instead of using handcrafted methods for feature extraction Convolutional Neural Network (CNN) was also used. Zeng et al. refer to CNN's advantages for HAR applied three-dimensional accelerometer data to CNN and classified with SoftMax layer at the end of the ANN [28] as shown in Figure 1.3. In the study where each accelerometer dimension is passed through a different convolutional lane, 96.8% success rate is achieved with Actitracker [36] dataset.

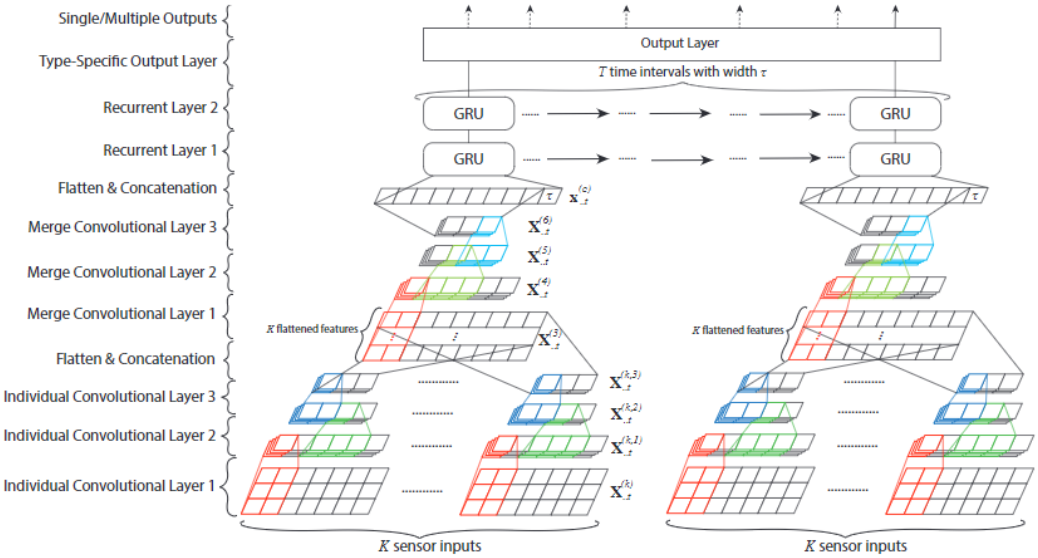
Moreover, as shown in Figure 1.4, a different architecture was offered by Panwar et al. in 2017 [37]. Those who used the SoftMax layer for classification also noticed that more complex data needs more convolution layers.



Source: [37]

Figure 1.4: CNN Architecture Offered by Panwar et al.

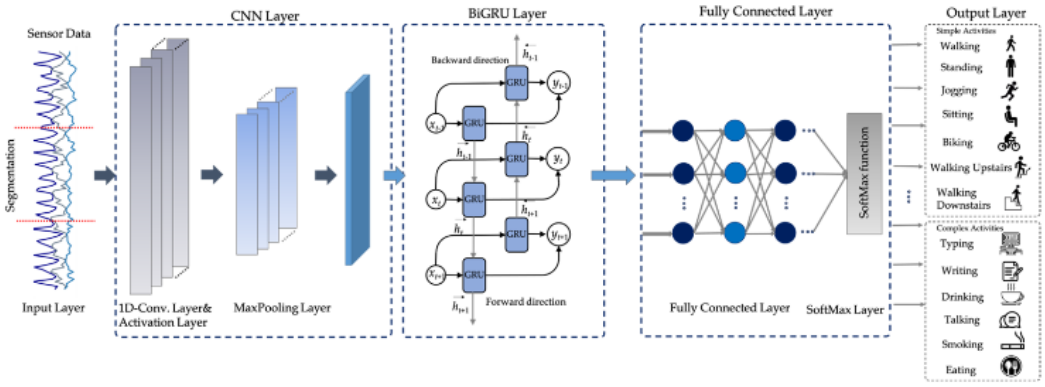
Due to the time series classification specialty of Recurrent Neural Networks (RNN), they have also started to use them in HAR applications.



Source: [23]

Figure 1.5: DeepSense Framework Architecture

In 2017, Yao et al. applied the architecture as shown in Figure 1.5 [23], which has convolutional layers for feature extraction, two-level Gated Recurrent Unit layers and SoftMax layer for classification. This architecture named with *DeepSense* achieved 95% F1-score on Heterogeneous Human Activity Recognition.

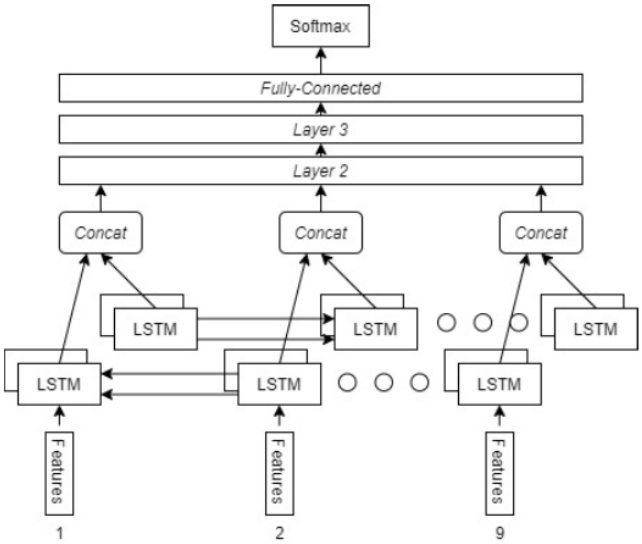


Source: [21]

Figure 1.6: Architecture Offered by Mekruksavanich and Jitpattanakul

In 2021, Mekruksavanich and Jitpattanakul offered the CNN-BiGRU model [21], as shown in Figure 1.6, and they reached average 0.974 F1-Score only using accelerometer data of the UTwente dataset [38].

Unlike the work of Mekruksavanich and Jitpattanakul above, in 2021, Nunavath et al. offered an RNN-only model for classification [39]. Their method achieved 98.75% accuracy for basic activities and 96.52% accuracy for specific activities on the UCI HAR dataset [40] using 10 seconds length sliding windows.



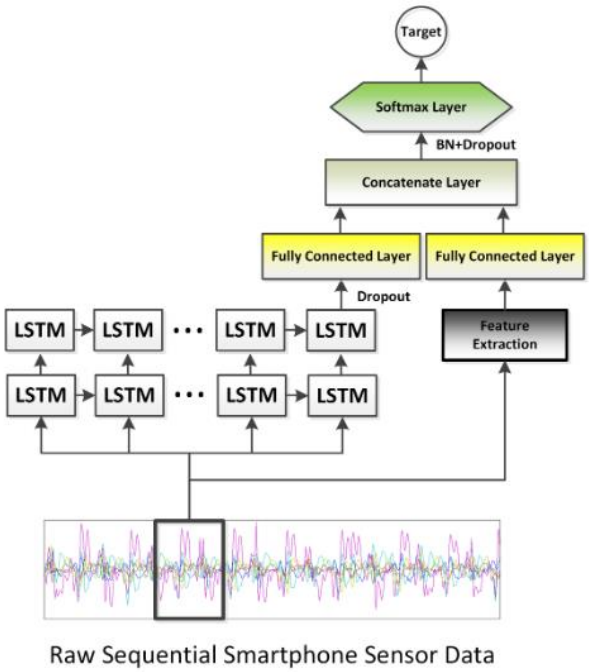
Source: [41]

Figure 1.7: LSTM Architecture Offered by Tarasevičius and Serackis

Similarly, in 2020 Tarasevičius and Serackis proposed a method with Long-Short Term Memory (LSTM) input blocks, as shown in Figure 1.7 [41]. This Bi-LSTM organized method is used to classify swimming style with accelerometer, gyroscope and magnetometer data collected from the wrist-worn sensor of swimmers. This model where the inputs are signal features reached 91.39% F1-Score.

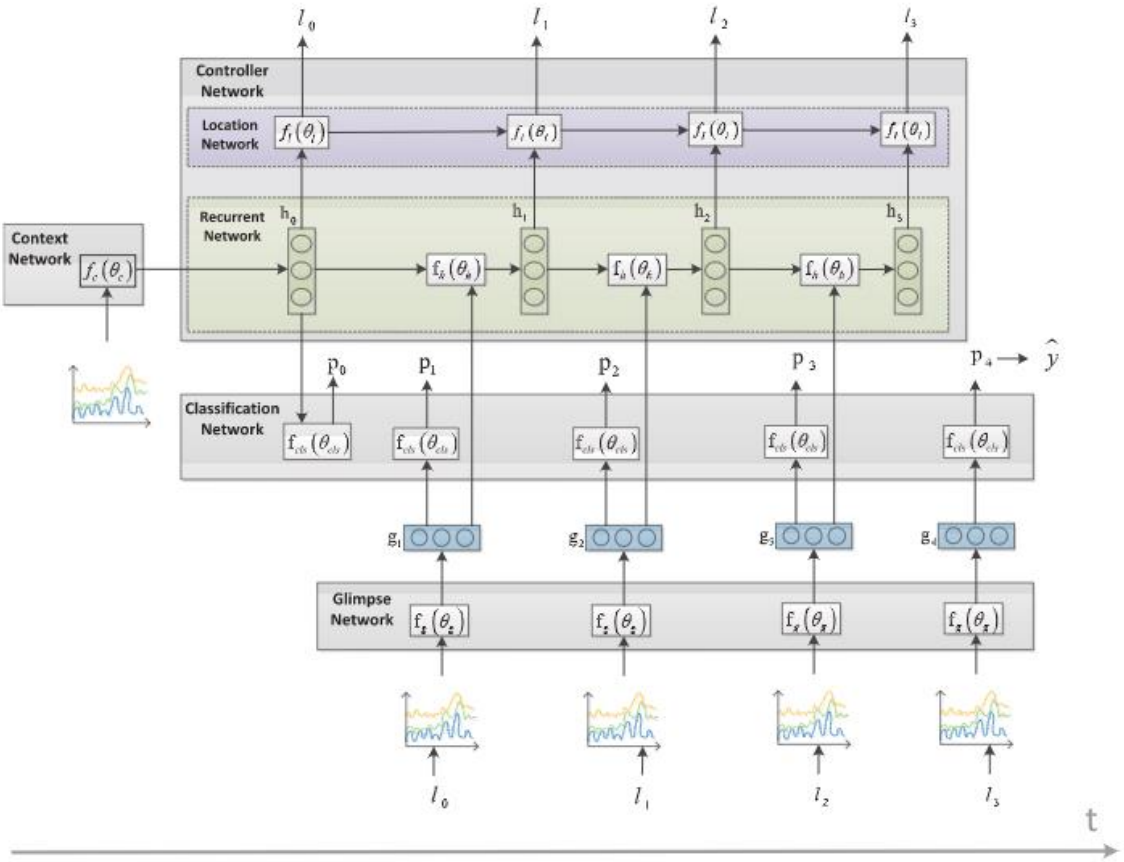
Same year, Chen et al. also offered the LSTM model [12], which has parallel fully connected layers for applying handcrafted signal features to the LSTM layer, as shown in Figure 1.8. Their fusion model reached 0.9644 accuracy rate. The *Maximum Full a Posteriori* approach derived from this model reached 0.9885 accuracy rate on the public dataset [40], which has wrist-mounted smartphone accelerometer and gyroscope data for human activity recognition.

In 2019 He et al. offered four levels of the network, as shown in Figure 1.9. Their model extracts the features with context vector and transfers them to the controller network built with location and recurrent network. Also, the glimpse network detects local features and transfers them to the classification network [27]. Their framework achieved 94.8% accuracy on the UCI HAR dataset and 95.35% accuracy rate in the *Weakly Labeled* dataset they have collected from the three-axis acceleration sensor of iPhone, which out on users trousers right pocket.



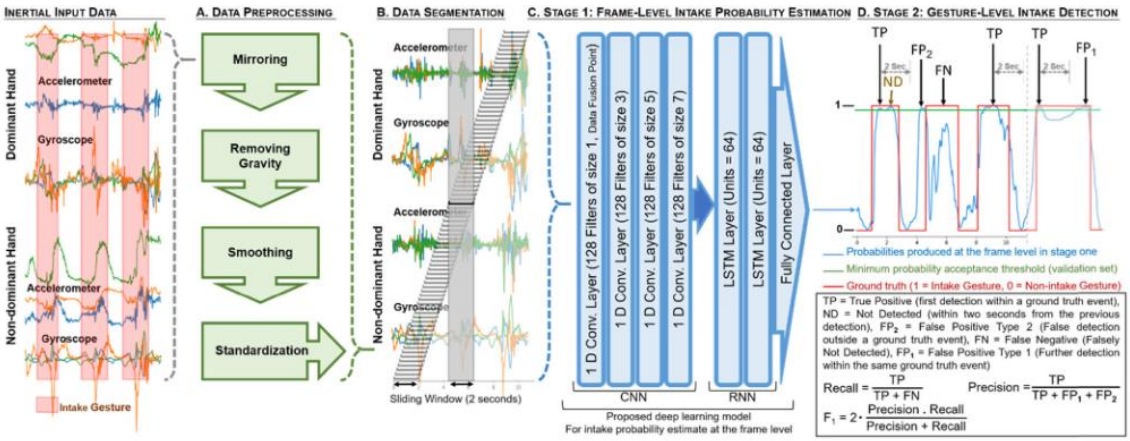
Source: [12]

Figure 1.8: LSTM Model Offered by Chen et al.



Source: [27]

Figure 1.9: Framework Offered by He et al.

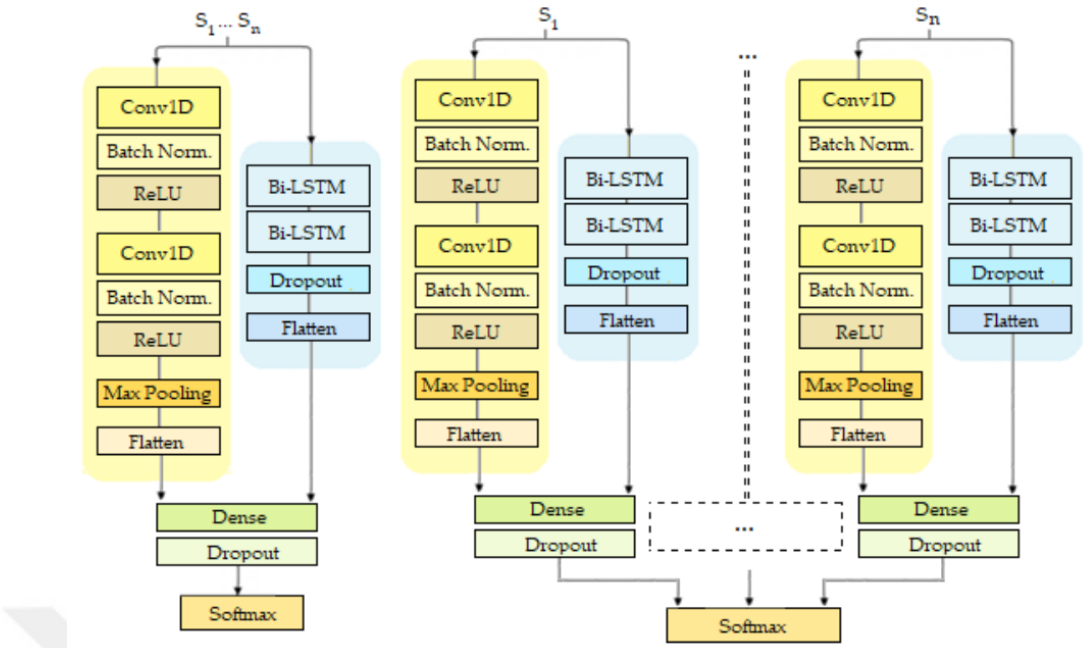


Source: [22]

Figure 1.10: HAR Model of Heydarian et al.

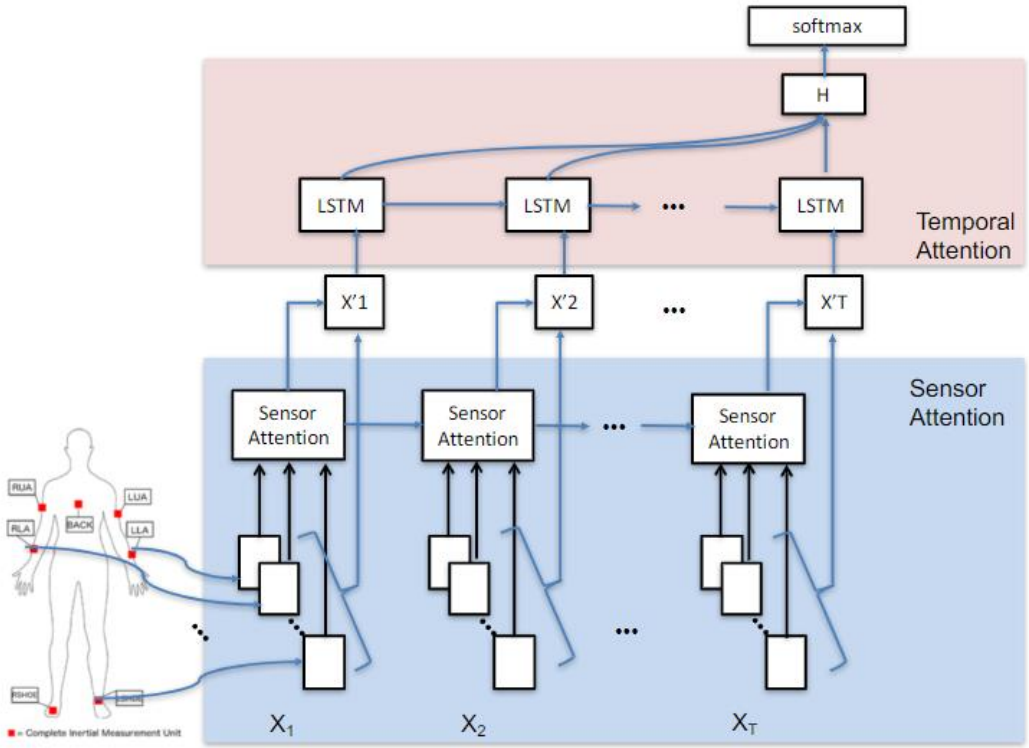
In 2020, Heydarian et al. proposed a model as shown in Figure 1.10 using CNN and LSTM [22]. Unlike the models offered above, they used accelerometer and gyroscope data taken from sensors worn on the right and left wrist and removed gravity effect additionally at the preprocessing phase. Then they reached 0.77 F1-Score with this method.

Same year, Kiprijanovska et al. used their smartwatch data collected from the accelerometer, gyroscope, magnetometer and rotation vector sensor [24]. Their work to detect gait abnormalities for falling risk proposed two different approaches, as shown in Figure 1.11, where the left-hand side data level network and the right-hand side decision level networks are represented. They reached an 83.7% accuracy rate using only accelerometer data and fusion of both decision and data level networks. However, their model showed maximum performance using accelerometer, gyroscope and rotation vector sensor combination with 88.9% accuracy rate.



Source: [24]

Figure 1.11: The Model Offered by Kiprijanovska et al.

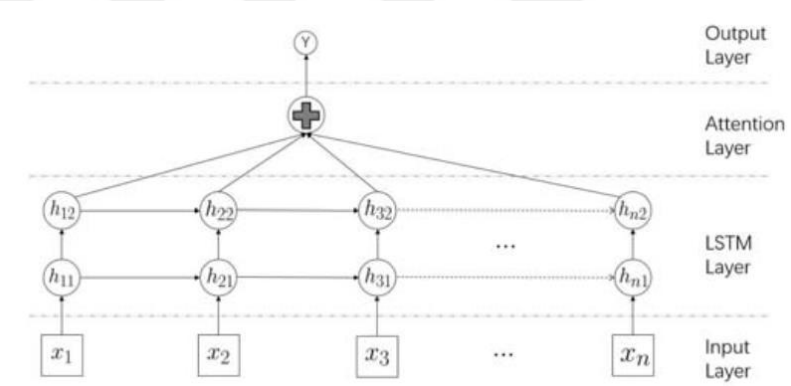


Source: [33]

Figure 1.12: The Model Proposed by Zeng et al.

In 2018, Zeng et al. proposed a sequential model based on attention mechanism [33] as shown in Figure 1.12, which has at the end of the model, in the temporal attention phase, they aimed to teach the network whole signal instead of a single piece. Moreover, in the sensor attention phase, they aimed to catch all critical modalities of sensor input with attention blocks. They tested their models with three different datasets and achieved the best F1-score results using sensor attention and temporal attention on the PAMAP2 dataset [42] with 0.899, but temporal attention without the Skoda dataset [43] sensor attention was achieved as 0.938.

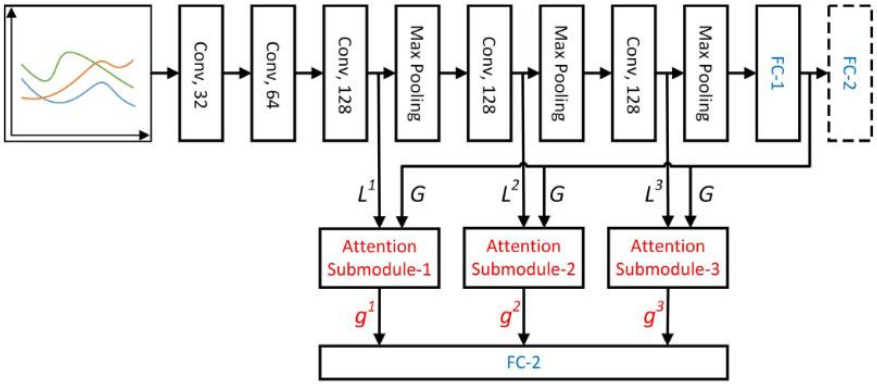
In 2019, Sun et al. proposed the Attention Based LSTM Network model as represented in Figure 1.13 [31]. Sliding data windows at the input layer applied to the LSTM layer for getting high-level features is aimed. Also, at the attention layer, learning inner relations is aimed. At the end of the model, SoftMax classifier is used and reaches 90.9% F1-score



Source: [31]

Figure 1.13: Model Offered by Sun et al.

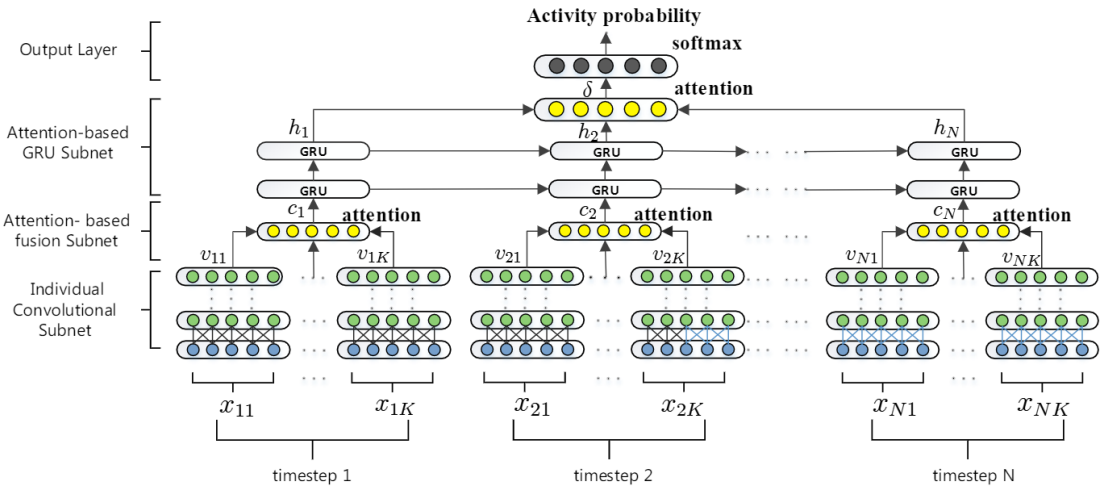
The same year, Wang et al. worked with UCI HAR and Weakly Labeled datasets. They offered a model with parallel-connected attentions blocks to the convolutional pipeline where each output has a ReLU function, as shown in Figure 1.14. Their model, which also has a fully connected layer, used SoftMax classifier at the end achieved 93.41% accuracy on the UCI HAR dataset when they used two attention blocks with parameterized compatibility and tanh function at the end of the block. For the Weakly Labeled dataset, the same conditions but three attention blocks achieved 93.83% accuracy.



Source: [17]

Figure 1.14: Model Architecture Proposed by Wang et al.

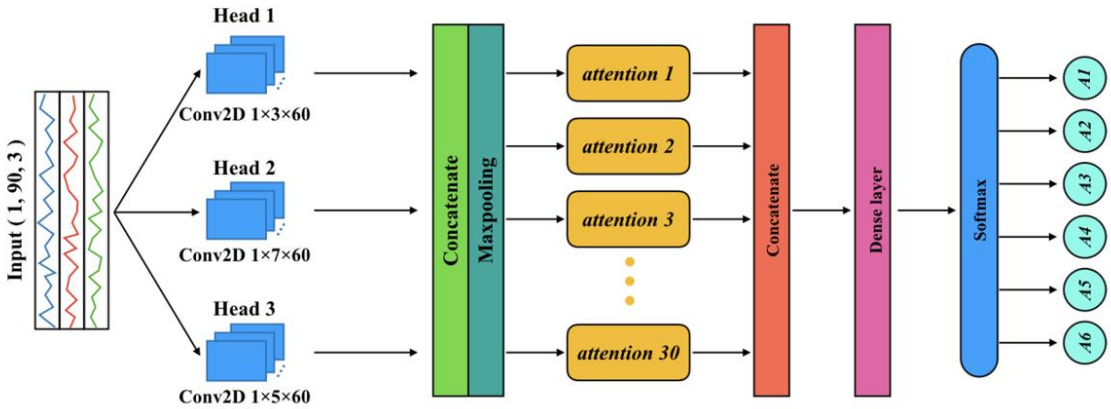
Ma et al. proposed *AttnSense* [18]. As shown in Figure 1.15, their model trained with in addition to the test data, Gaussian Noise added augmented data, and Fast Fourier Transform (FFT) applied frequency spectrum data. Batch normalization is applied at the Individual Convolutional Subnet in the model to reduce covariance shifting. At *Attention-Based Fusion Subnet,* self-attention is determined, and at the *Attention-Based GRU Subnet,* GRU's are used for determining long-term dependencies instead of LSTM due to its complexity. The model has fully connected SoftMax function-based classification layer at the end, achieved F1-scores as 0.965 for Heterogenous [44], 0.931 for Skoda and 0.893 for PAMAP2 datasets.



Source: [18]

Figure 1.15: AttnSense Model

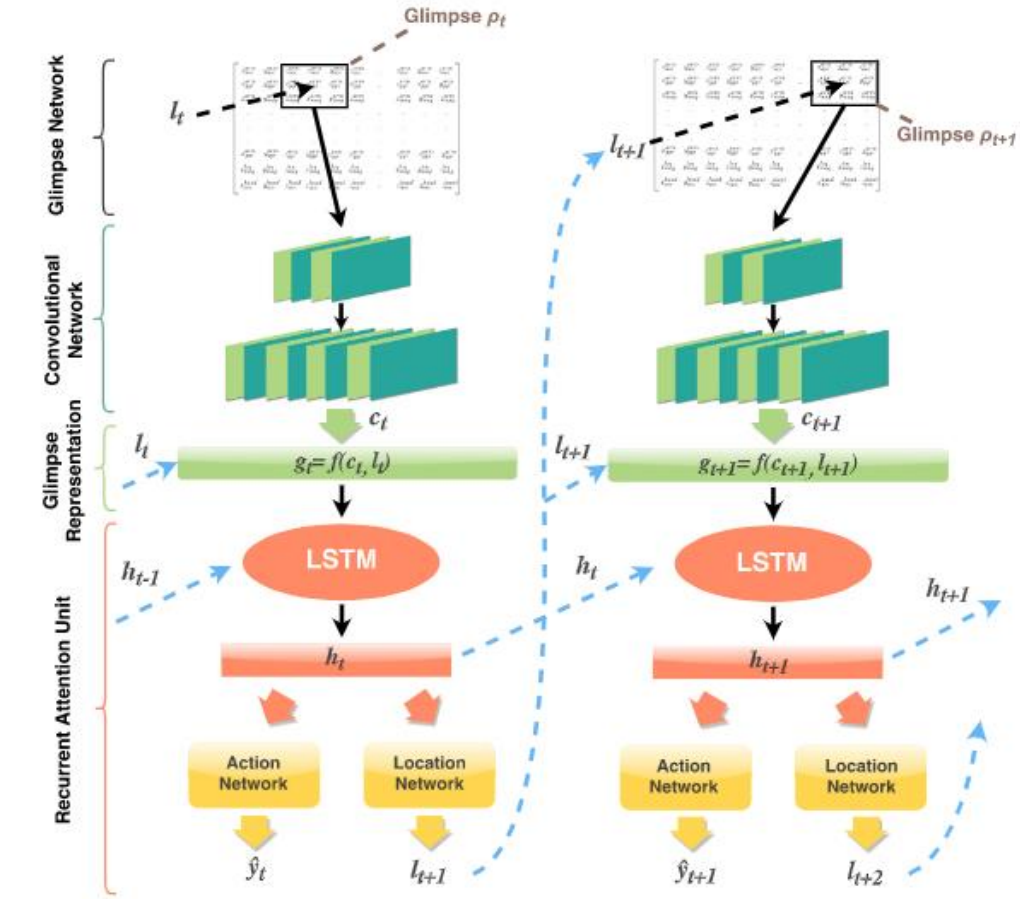
Zhang et al. proposed another attention-based network as given in Figure 1.16 [14]. They filtered noisy data at the input and segmented it with specific window sizes. Three-dimensional input data is applied to a multi-headed CNN layer and concatenated at the end. After the parallel attention layer at the end of concatenating, the SoftMax classifier was applied. This multi-head attention approach achieved 0.954 F-measure on the WISDM dataset.



Source: [14]

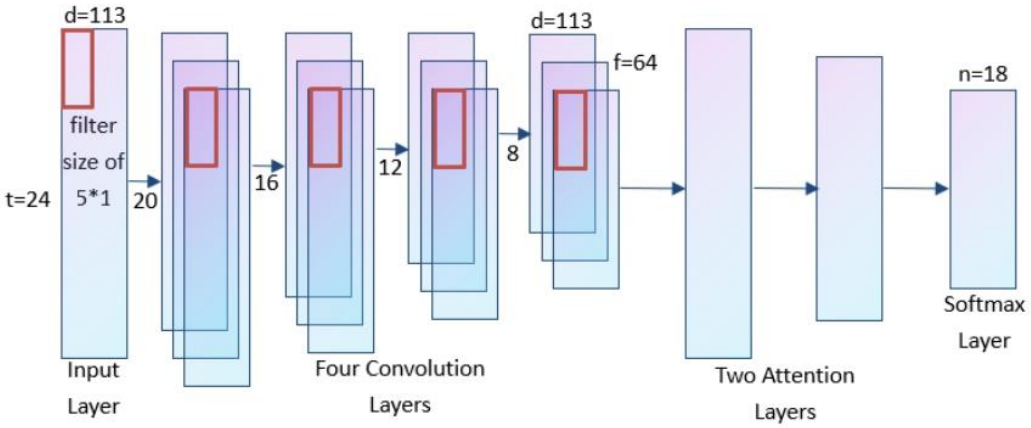
Figure 1.16: Multi-head Attention Network Proposed by Zhang et al.

Chen et al. proposed a semi-supervised network for HAR applications for training with labeled and unlabeled data together [15]. As shown in Figure 1.17, their offering for imbalanced data type distribution aimed to classify multiple activities. At the Glimpse Network, they aligned input sensor data to matrices for keeping the relation between features and at the Convolutional Network, they aimed to extract high-level features. Then after Glimpse Representation Layer, which performs linear transformation, at the Recurrent Attention Unit, LSTM sets the relation between time sequences up and transferred information to Action and Location subnetworks. Location network feeds data back, and Action network tries to predict activity label. They also used the Partially Observable Markov Decision Process (POMDP) technique to overcome training problems. This network named with Recurrent Attention Model (RAM) has shown 0.9425 for MHEALTH [45] with 13000 labeled samples, 0.8342 for PAMAP2 with 20000 labeled samples, 0.8184 for UCI HAR with 13000 labeled samples.



Source: [15]

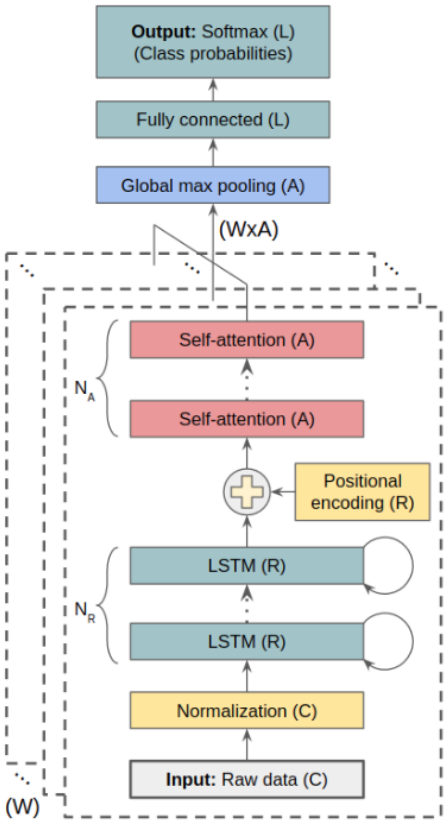
Figure 1.17: The Proposed Model of Chen et al.



Source: [25]

Figure 1.18: DeepConvAttn Model

Zhang et al. proposed an attention network [25] with modifying DeepConvLSTM [46] network and named with DeepConvAttn as shown in Figure 1.18. The model designed to eliminate parallel computing problems of LSTM with replacing Attention layers achieved 0.928 F1-score for gestures and 0.906 F1-score for locomotives using the Opportunity dataset [47].

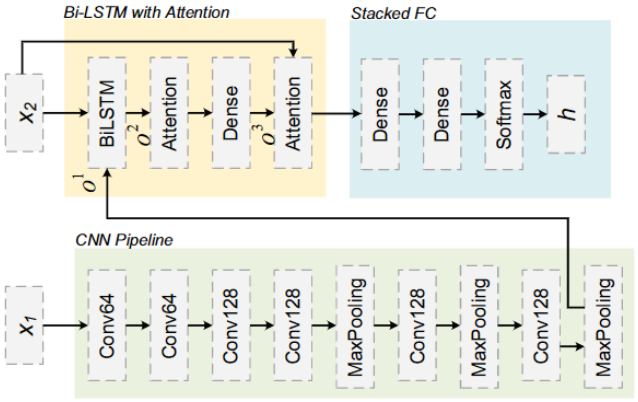


Source: [32]

Figure 1.19: Model Proposed by Betancourt et al.

Betancourt et al.applied sliding windows of UCI HAR to their model [32] as illustrated in Figure 1.19, where N_A, self-attention layer size is 20 and N_R, recurrent layer size, is 3. They also used positional encoding for identifying the most appropriate time steps. With and without positional encoding, their two approaches achieved 97.1% and 94.1% accuracy, respectively.

Wang and Zhu used the UCI HAR dataset on a hybrid approach [26], shown in Figure 1.20. After denoising input data, handcrafted signal features were applied to the Bi-LSTM Attention Layer in addition to the created output by the CNN pipeline and achieved 95.58% accuracy.



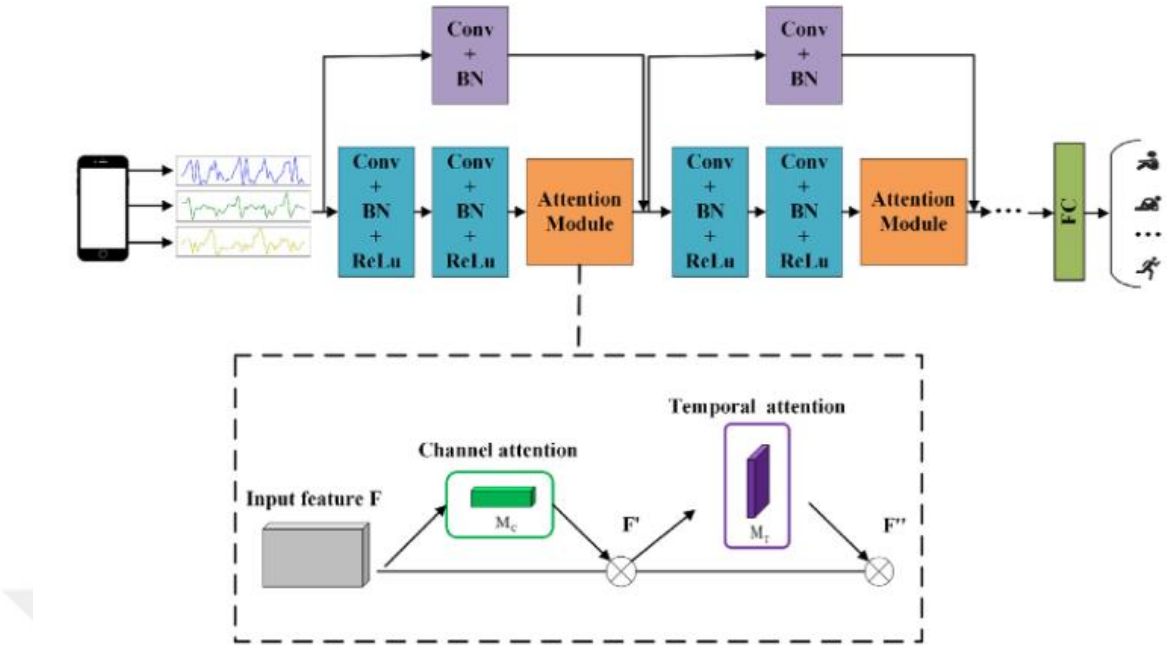
Source: [26]

Figure 1.20: Hybrid Approach by Wang and Zhu

In 2021, Buffelli and Vandin replaced the GRU layer of DeepSense with the Attention layer for defining temporal dependencies and named their model with TrASenD [30]. In addition to TrASenD, the TrASenD-BD method used substitution of attention block with bidirectional-RNN offered by Schuster and Paliwal [48] achieved 0.798 for HHAR [44], 0.650 for PAMAP2, and 0.681 for USC-HAD [49] F1 scores. TrASenD-CA model where the GRU layer used with attention mechanism like Xu et al. [50] achieved 0.797 for HHAR 0.659 for PAMAP2 and 0.687 for USC-HAD F1 scores.

However, TrASenD achieved 0.848 for HHAR, 0.723 for PAMAP2, and 0.702 for USC-HAD F1 scores. They improved their models with users' feedback also. This personalized experiment increased the performance of TrASenD, which has maximum accuracy to 0.889 for HHAR, 0.749 for PAMAP2 and 0.759 USC-HAD.

Gao et al. proposed a sequential dual attention model named Dual Attention Network (DanHAR) [20] for combining temporal and channel attention as shown in Figure 1.21, where BN is Batch Normalization. Their two-dimensional convolution-based channel attention module is used for determining input features. Max pooling technique in this layer provided channel-wise attention finer. Moreover, six convolutional layers with a short-cut connection for adding residual mapping are used. In contrast, two concatenated pooled features are convolved at the temporal attention layer, multiple sensors are unified, and information is saved. In the end, their approach boosted feature discrimination. They reached test accuracy of 98.85% for WISDM, 79.03% for UNIMIB SHAR, 93.16% for PAMAP2, 82.75% for OPPORTUNITY, and 94.86% for Weakly Labeled datasets.

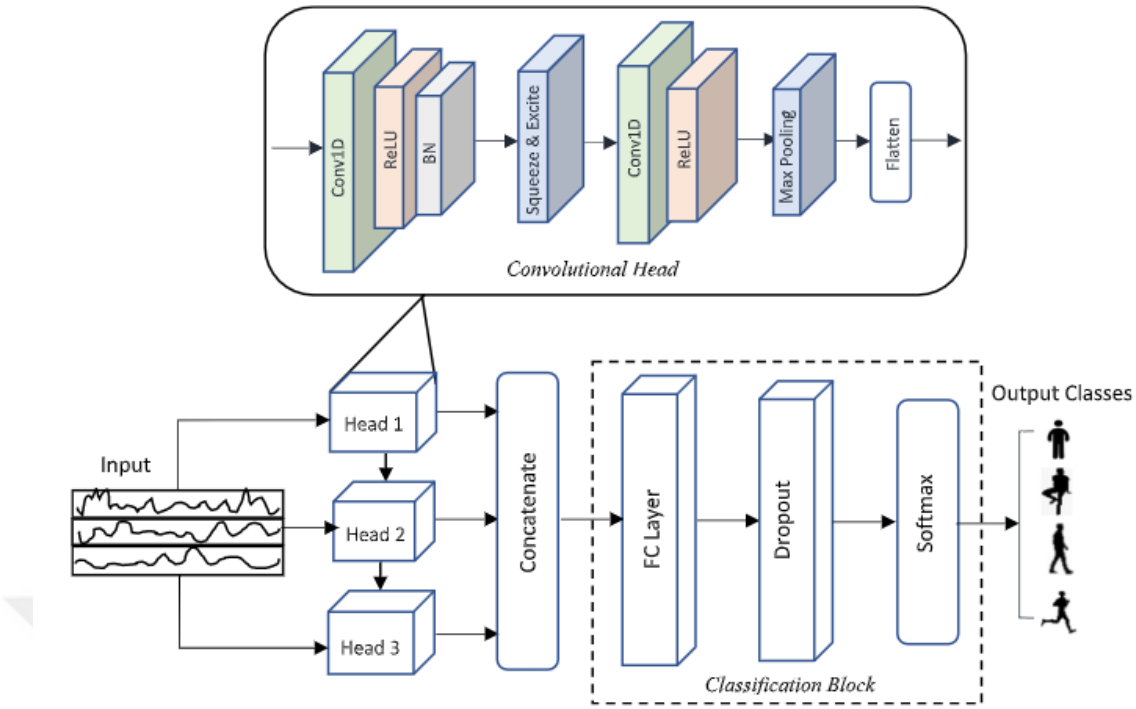


Source: [20]

Figure 1.21: DanHar Model

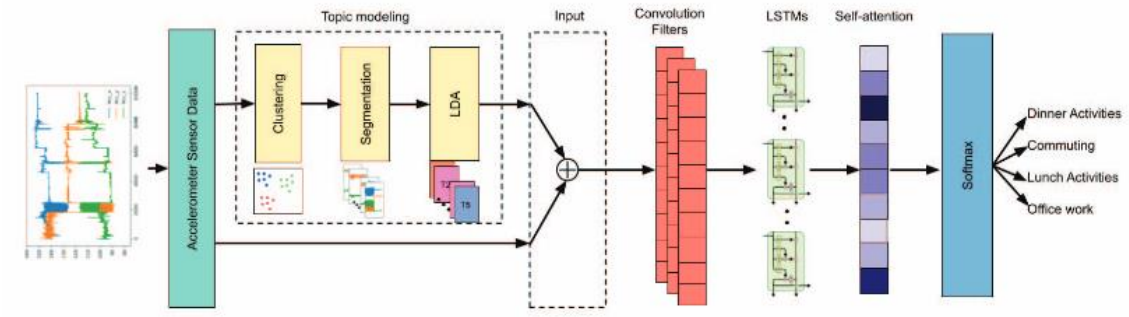
Khan and Ahmad worked with fixed-size windows of normalized data [16]. Their model has a one-dimensional convolutional layer for crafting input features. After the Convolutional layer, a non-linear ReLU function is used for reducing the vanishing gradient problem.

As shown in Figure 22, Multi-head CNN architecture improved the classification results. Also, the squeeze & extraction [51] module provides an attention mechanism. The proposed model achieved 0.9818 accuracy performance on the WISDM dataset and 0.9538 on the UCI HAR dataset.



Source: [16]

Figure 1.22: Multi-head CNN Offered by Khan and Ahmad

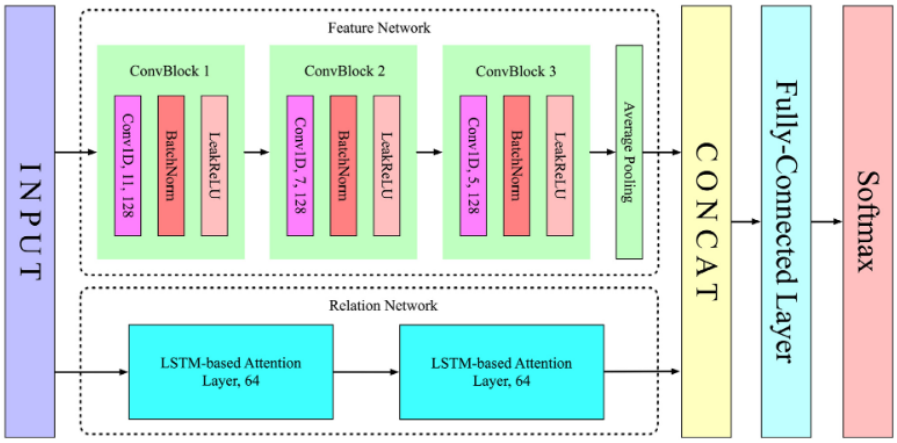


Source: [19]

Figure 1.23: CARTMAN Model

Chandrasekaran et al. proposed a model for classifying complex activities named CARTMAN,s shown in Figure 1.23 [19]. Their model used the Latent Dirichlet Allocation (LDA) [52] model to extract features. The input data is clustered with k-Means Clustering for segmentation, and 5 minutes length windows are passed through the LDA layer. LDA modeled data and pure sensor data channels are passed through DeepConvLSTM for using its self-attention mechanism. This CARTMAN approach used the Ubicomp 08 Complex Activity dataset [53] and achieved a 0.95 F1-score weighted average.

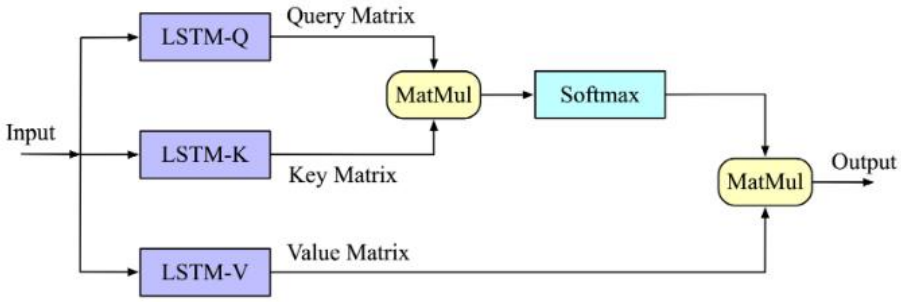
Xiao et al. developed a model with parallel convolutional and LSTM pipes, as shown in Figure 1.24, named with Perceptive Extraction Network (PEN). The PEN has a parallel feature network for local feature extraction and a relation network for sensing global input patterns.[13]



Source: [13]

Figure 1.24: Architecture of PEN

In Figure 25, the LSTM-based Attention Layer is represented used in the Relation Network of PEN. This parallel network layer to Feature Network Layer improves multivariate time series classification results. As a result, the PEN model achieved F1-scores, 98.97 for WISDM, 96.33 for UCI_HAR, 97.78 for PAMAP2, and 96.89 for OPPORTUNITY.



Source: [13]

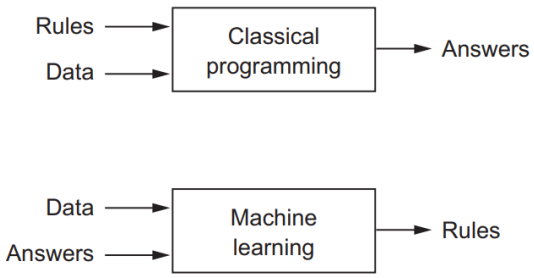
Figure 1.25: Relation Network of Pen

1.3 BACKGROUND & THEORY

1.3.1 Artificial Intelligence & Machine Learning

Artificial Intelligence (AI) concept was born in the 50s. With the increasing computing power of hardware and the number of datasets in different areas, machine learning and deep learning methods are gained popularity for different reasons such as classification, recommendation and prediction operations.

While Artificial Intelligence refers to mimicking human behaviors, Machine Learning (ML), a part of the AI, is specified for changing decisions concerning past data provided to the system. The system referred to here is mentioned as Machine Learning Paradigm by Chollet, as shown in Figure 26 [54]. While traditional algorithms designed for searching answers for questions try to find out system rules against input (data) and output (answers).



Source: [54]

Figure 1.26: Machine Learning Paradigm

Chollet also said that Machine Learning algorithms are generally performed with statistical approaches, specified ways of ML, and deep learning (DL) methods have one or more layers used for the empirical learning process.

In ML and DL methods, a model is created then data and its answers are related. This process is calling as *training*. While the training process, the model learns how to process and answer input data. After the training phase, the model will respond to the input data.

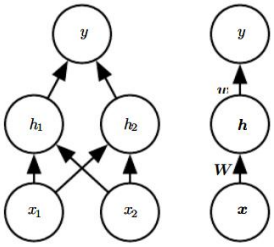
1.3.2 Deep Learning

Deep Forward Network, Feedforward Neural Network, or Multilayer Perceptron(MLP) represent the same AI models [55].

For mathematically, they can be represented as illustrated in Equation 1.1, where x is input, $f^{(n)}$ is the function of the n'th layer of the network, and y is the mapped category of x. Each f function represents a hidden layer that learns how to use its inputs to reach the desired output.

$$Y=(f^{(n-1)}(\ldots f^{(2)}(f^{(1)}(x))\ldots)) \quad (1.1)$$

In Figure 1.27, a schematic view of a simple MLP with a single hidden layer containing two units is illustrated at the left-hand side and, a simplified representation is represented at the right-hand side.

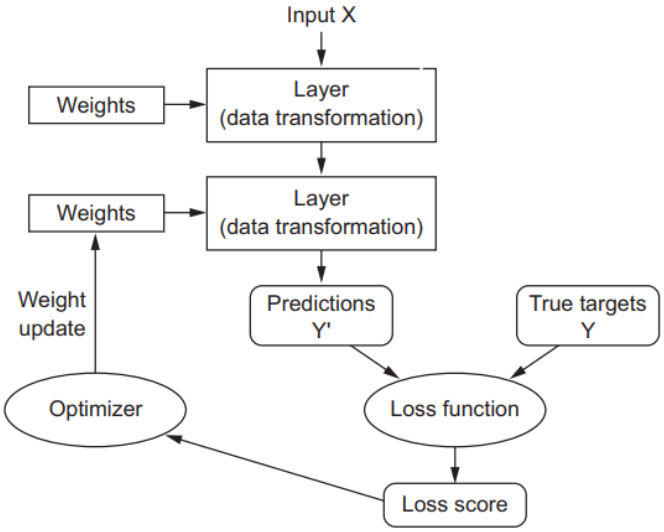


Source: [55]

Figure 1.27: Simple MLP Scheme

Chollet [54] summarized the anatomy of Neural Networks, *inputs*, *layers*, *loss function*, which refers to learning feedback function and *Optimizer* for determining learning proceeds. As shown in Figure 28, loss function and Optimizer can be defined for the simplified learning process. A differentiable loss function compares actual Feed-forward prediction results and accurate results during the learning process, then calculates a loss score. The iterative Gradient Descent algorithm is used for minimizing losses. Derivative of the differentiable loss function gives the minimum point of this function with finding where the derivative is zero. For neural networks, finding combinations of the smallest weights leads to finding the minor loss function.

The Optimizer is generally built with Stochastic Gradient Descent(SGD), which starts with a random initial value if not determined and found the slope of the loss function; in other words, it computes the gradient [55] and then updates the gradient until it is closest to zero. At the end of the SGD algorithm, layers' weights are updated for better results, so *Back Propagation* has happened.



Source: [54]

Figure 1.28: Simplified Learning Scheme

1.3.3 SoftMax Classifier

The SoftMax function, as defined in Equation 1.2, is generally used in the last part of the networks due to predicting probabilities in multinomial distribution over *n* different possible values. The SoftMax function output cannot change by adding a scalar to the input vector. As a result of this increase of the unit affects others as decreasing too.

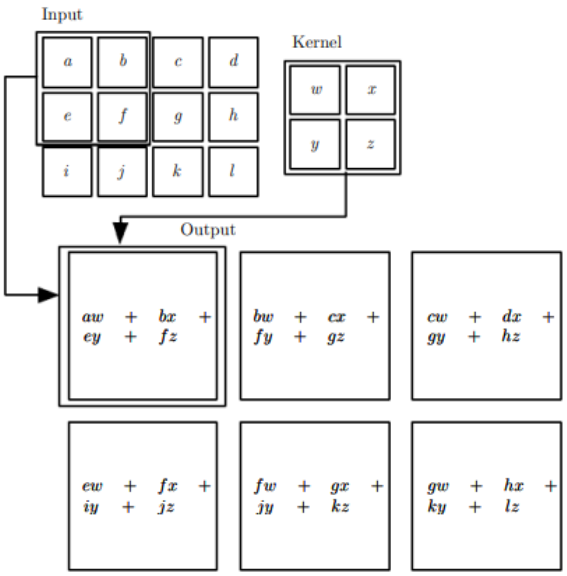
$$\mathrm{SoftMax}(x)_i = \frac{\exp(x_i)}{\sum_{j=i}^{n} \exp(x_j)} \tag{1.2}$$

1.3.4 Convolutional Neural Networks

Convolution Neural Networks (CNN) specializes in processing data as grids [55], where the convolution operation is applied at least in one of the layers. Convolution operation($*$) as given in Equation 1.3 is generally used for generating features at the output(s) from input data(x) with a kernel(w).

$$s(t)=(x * w)(t) \tag{1.3}$$

A simple convolution process can be performed in 2-D space as illustrated in Figure 1.29.

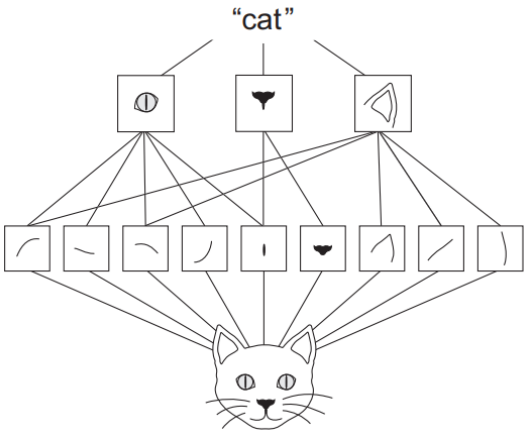


Source: [55]

Figure 1.29: Simple 2D Convolution Operation

In traditional neural networks, global patterns are extracted in layers, despite that in CNNs, convolution layers extracted local patterns. These extracted patterns can be recognized anywhere by convolutional layers [54].

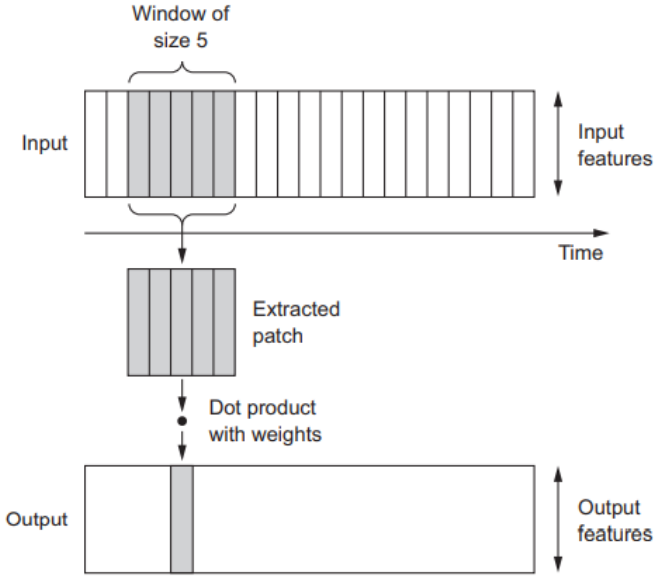
While adding more convolution layers to CNN, the network learns more complex features, as shown in Figure 1.30. However, adding more layers reduce the trainability of the model.



Source: [54]

Figure 1.30: CNN Spatial Hierarchy Example

The convolution process can be performed in single-dimensional space and two-dimensional space. A convolution process in a single-dimensional space can be exampled as represented in Figure 1.31, where the time is a spatial dimension. This figure shows that 1D convolution applies to the time series data, and the subsequence data is generated. This process reduces input length and creates local patterns for recognition after the model training phase.



Source : [54]

Figure 1.31: 1D Convolution Example

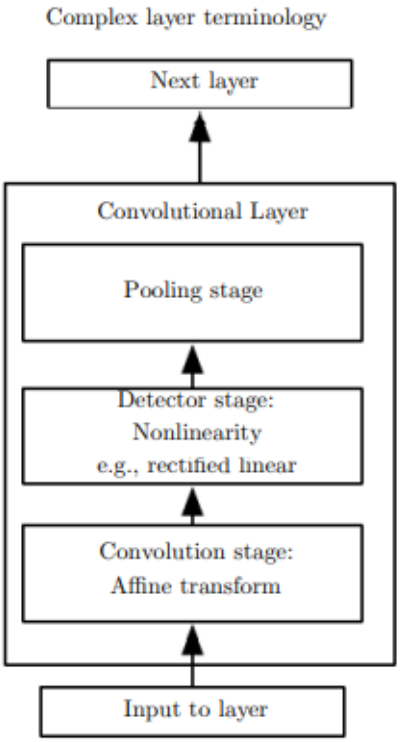
Goodfellow et al. pointed out three ideas of Convolution Neural Networks: sparse interactions, parameter sharing, and equivariant representations [55].

While in traditional neural network layers, each input and output interacted with each other, there is sparse interaction or lightweights in CNN due to the kernels making input smaller. So fewer parameters are stored, the efficiency of the network increases, and the memory requirements decrease.

In neural networks, each weight of a layer is used once when computing. However, in CNN, each kernel cell is used in every possible input with parameter sharing meta and provides learning more related features in one set instead of separate sets.

The parameter sharing attribute of CNN provides equivariance to the translation process; thus, output changes when the input change. Goodfellow et al. also pointed out the equivariance in time series with convolution [55].

Goodfellow et al. modeled CNN with three steps, as shown in Figure 1.32. At the first step, different convolution processes are performed parallel; at the second step, non-linear activation functions are performed, also called the *detector* stage. The pooling function is applied at the last step, and output is modified while summarizing nearby data cells for the following uses.

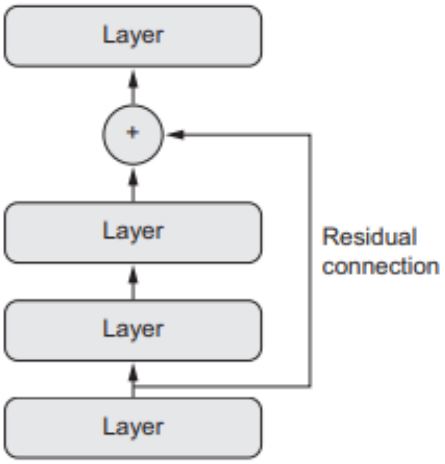


Source: [55]

Figure 1.32: The CNN Model Referred by Goodfellow et al.

Convolution layers are generally good at generating features for short-time dependencies. Although their performance can be increased with Max Pooling, a residual connection represented in Figure 1.33 transfers previous downstream data to a later next layer for maintaining dependencies. In addition to Max Pooling and residual connection methods, graph-based networks such as RNNs can also be used.

Like residual connections, Batch Normalization can prepare deeper network models. The data was maintained batch-wise with an exponential moving average during training and increased gradient propagation performance.



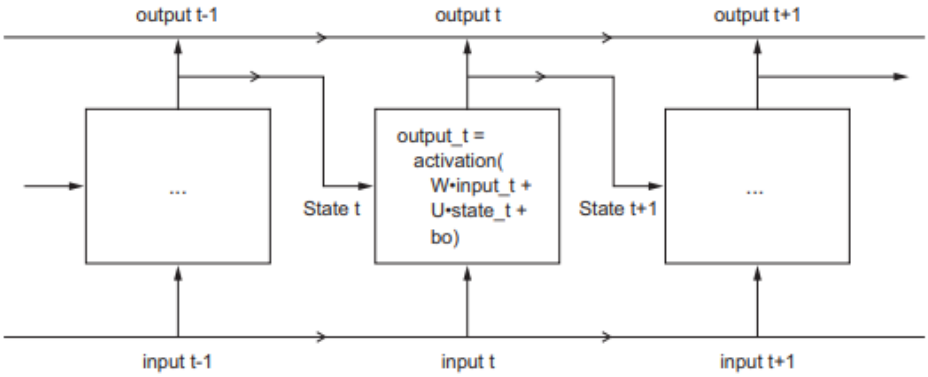
Source: [54]

Figure 1.33: Residual Connection

1.3.5 Recurrent Neural Networks

When1D Convolution Networks are investigated on time series, and the network shares parameters across the time; however, this is shallowed. Although the parameters sharing method of CNN, in Recurrent Neural Networks (RNN) internal loop feedback to the output of the layer to its input.

Figure 1.34 shows that simple RNN application over time. The current time is represented with the t value. The output of the RNN block is held and directed to the RNN block at t+1 time. So the state is covered and transferred. Thus each state memory of the previous block is transmitted to the future while the network is working. However, long-term dependencies are missing in RNNs.



Source: [54]

Figure 1.34: Simple RNN Application Over Time

1.3.5.1 Vanishing Gradient

For Feedforward Neural Networks, adding more layers makes the network untrainable [54]. In contrast, RNNs are good at learning for short periods. Untrainability occurs with Vanishing Gradient Problem as happened in Feedforward Networks. The feedback data provided with the Back Propagation algorithm become weak or completely lost deeper or longer propagation, then the network becomes untrainable.

1.3.5.2 Lstm & Gru

Hochreiter and Schmidhuber developed the Long Short-Term Memory (LSTM) algorithm to solve The Vanishing Gradient Problem [56]. In Figure 1.35, a block diagram of the LSTM is represented. In the given figure, forget gate determines what to forget at the last data, the input gate determines what will be written to the internal cell state, and the output gate is responsible for extracting features for the output of the cell whether the block is used in the last sequence or not.

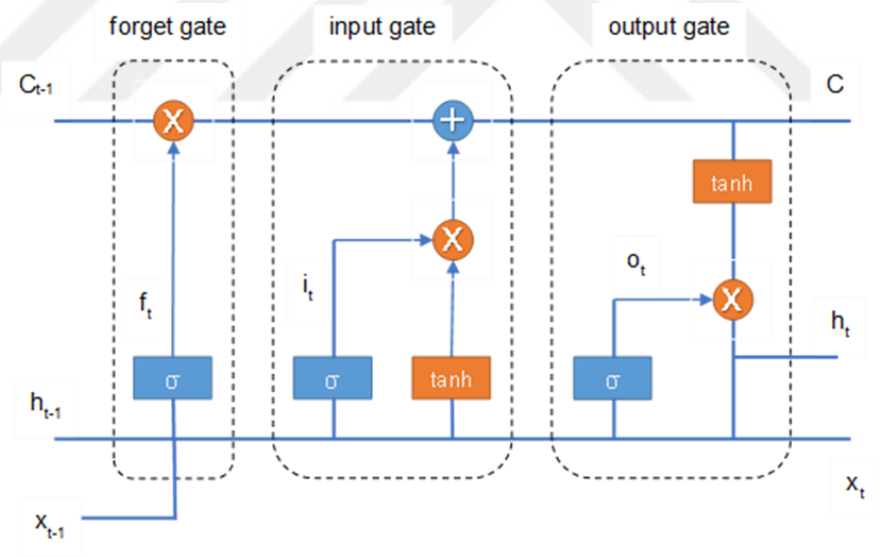


Figure 1.35: LSTM Block Diagram

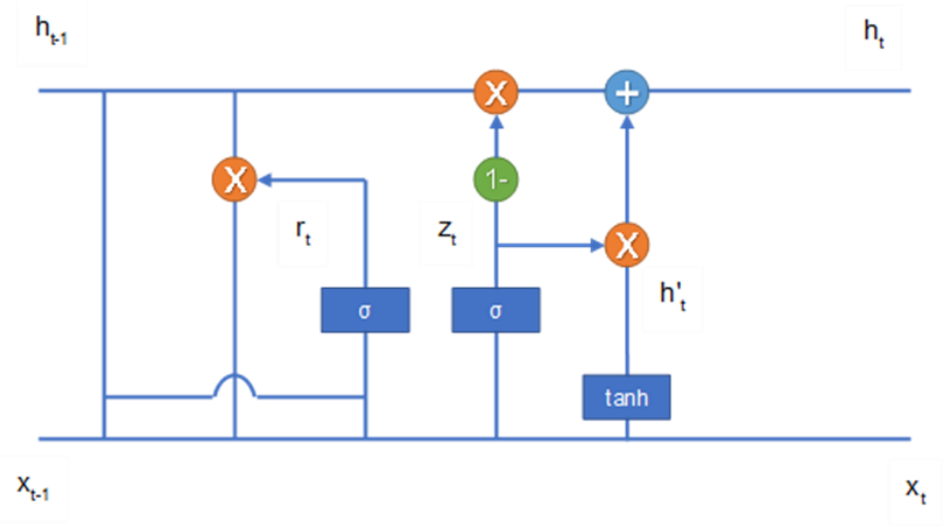


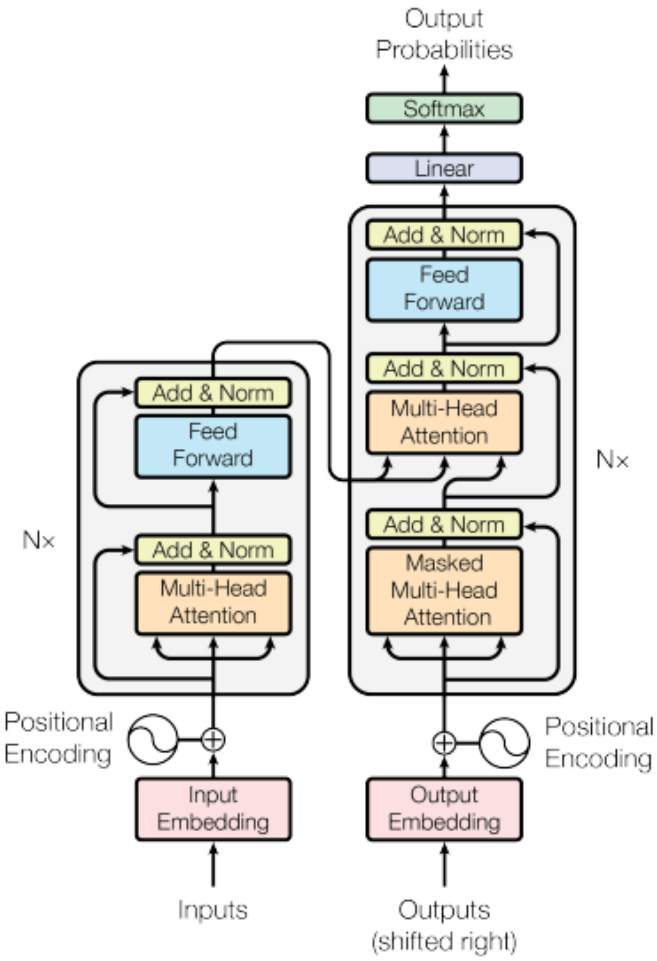
Figure 1.36: GRU Model

Gated Recurrent Unit (GRU), one of the LSTM variations, does not need to maintain an internal state instead of the original LSTM. The information stored in the internal state is wide into the hidden state of GRU.

As shown in Figure 1.36, there are also three gates in GRU Model as like LSTM. Reset gate(r_t) at the first phase, a combination of input and forgets gate of LSTM, determines how much memory will be forgotten and how much information will be passed from the previous block. Update gate (z_t) decides how much data will be transferred to the next block, like the output gate of LSTM. Furthermore, the Current Memory Gate is responsible for what will be forgotten from the previous block's data at the last phase.

1.3.6 Transformers

In 2017 Vaswani et al. proposed a Transformer model [34], as illustrated in Figure 1.37. Their sequence to sequence model is built with an encoder and decoder structure. In their autoregression model, While the encoder maps the input sequence, the decoder generates an output sequence both they have multi-head self-attention layers. Their work named “Attention is All You Need” described the attention function with mapping queries with related key and value paired vectors. The output of the attention function is generated with the summing of values computed by the compatibility function of the query and its corresponding key.



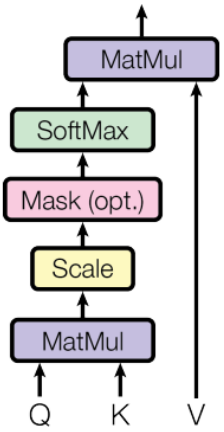
Source: [34]

Figure 1.37: Transformer Model

1.3.6.1 Scaled Dot-Product Attention

In Scaled Dot-Product Attention, as given in Figure 1.38, the attention is calculated with a formula illustrated in Equation 1.4. The dot product is applied to Query(Q) and Key(K) matrices, and the result is scaled with the square root of the Key matrix's dimension. Then Softmax function is applied for defining weight values. Vaswani et al. pointed out that attention of queries is computed and packed with query matrix simultaneously and packing keys and Values(V) together.

$$\text{Attention}(Q,K,V)=\text{SoftMax}(\frac{QK^T}{\sqrt{d_K}})V \quad (1.4)$$



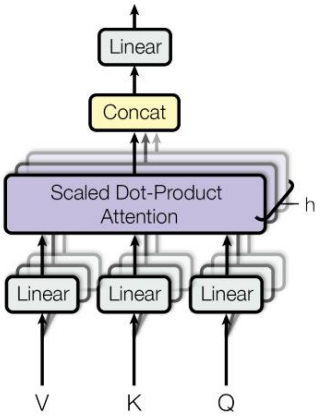
Source: [34]

Figure 1.38: Scaled Dot Product Attention

Unlike the additive attention offered by Bahdanau et al. [57], dot-product attention is faster and more efficient for matrix multiplication [34].

1.3.6.2 Multi-Head Attention

Vaswani et al. applied their attention mechanism for linearly projected queries parallel for extracting essential information from different subspaces. These extracted information concentrates these attention layers' results, as illustrated in Figure 1.39. They also pointed out that reducing dimension in attention heads provides similar computational costs like single attention used fully dimensional data.



Source: [34]

Figure 1.39: Multihead Attention

1.3.6.3 Advantages of Attention

Advantages of self-attention can be summarized concerning the work of Vaswani et al. [34]

- Keys, Queries and Values are coming from the same place so that each position can be related to previous positions, in other words, all positions.
- Scaled dot-product attention by masking values at the input for preventing unwanted connections.
- Attention computation can be parallelized, and this operation shows the exact cost as fully dimensional single attention
- Long-term dependencies can be learned with shorter paths.
- Self-attention computations are faster than recurrent layers, so they can be trained faster.
- The convolution operation is expensive when compared with the recurrent process. However, Self-Attention and pointwise Neural networks have the same complexity and are faster.

1.4 CONTRIBUTION OF THESIS

Although there are only accelerometer data used works, unlike many other works used combination of different wearable sensor data. In this thesis, only accelerometer data is used. The proposed method shows its performance on the HANDY dataset for classifying the person's activities and the person.

Concerning the proposed model by different researchers as mentioned above in the Previous Work section, a lightweight model is aimed in this thesis. 3 Convolution Layers are used for feature extraction, and a single Multi-Head Attention Layer is used for relating long-term dependencies in the time series data provided at the input stage. The model finalized with SoftMax classifier ended Neural Network.

The proposed model gave more successful results than the methods proposed by Açıcı et al. Instead of LSTM or RNN, the attention mechanism used in this work showed faster performance.

CHAPTER II

METHODS

In this thesis, the HANDY dataset collected by Açıcı et al. was used for classifying participants' activities. The HANDY dataset has different sensor data, which is explained in detail in the next chapter. However, only accelerometer data is used in this work due to the advantages of only-accelerometer data.

Sampled accelerometer data is positioned in the participant's wrist divided into fixed-length sequence chunks for training and testing.

There are some assumptions are made for experiencing the HAR process with HANDY in this thesis:

- Nine different activities are evaluated, and 4 seconds length windows are used for each activity for every unique person are used as input. It is thought that all unique features for activities in a length of 4 seconds can be determined.
- Not all signal data for an activity and the participant who performs it are used. It is assumed that periodic movements have occurred during all the collecting phases. Despite that, a part of the collection has to be enough.
- In the dataset, data rows whose data is missing are ignored. Due to the periodicity of the activity. It is assumed that ignored data row can be maintained during the training phase with its periodic representation due to the long-term dependency performance of attention layers.
- As mentioned before, accelerometer data is noisy. Another assumption is made regarding ignoring the confusing effect of noise in the CNN layer.

2.1 MODEL ARCHITECTURE

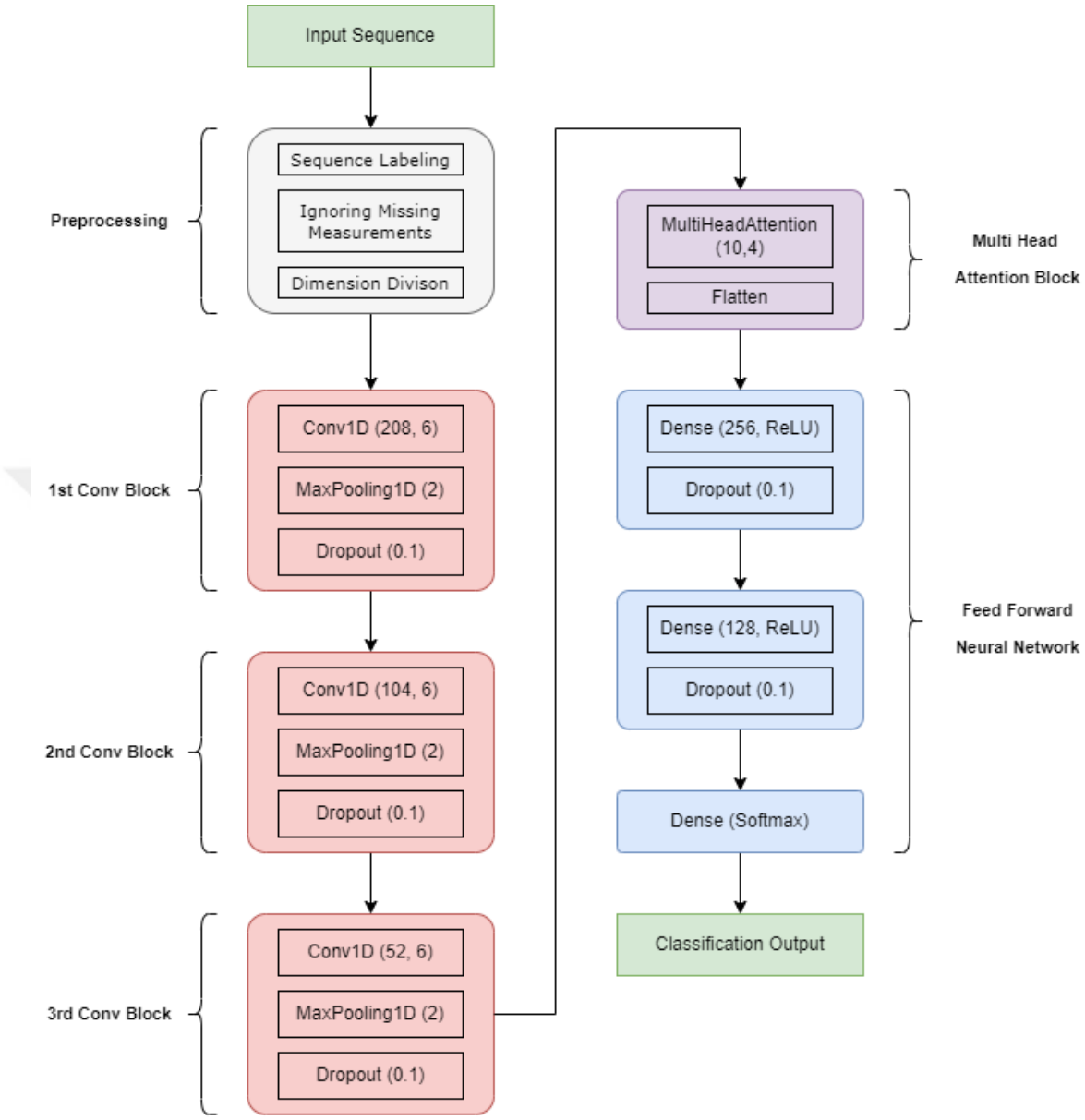


Figure 2.1: Proposed Model

2.2 PREPROCESSING

As mentioned before, at the preprocessing step in training, input data is divided into sub-segments, and these segments are labeled according to the activity type or person they belong to. Low noise accelerometer data is extracted with x, y and z axes in the HANDY dataset. Also, in case of missing data, in this step, missing data is ignored. Relying upon the periodicity of input data and long-term maintaining specification of attention layer, missing data sequences are ignored.

Four seconds length subsequences are derived from the chunk data. It means that 208 samples were derived from multiplying of 4 seconds and 52Hz sample rate. The input data is augmented with a 208 sample length sliding sample window one by one sample.

2.3 CONVOLUTION LAYER

The proposed model has three levels of one-dimensional convolutional layers. From the first to last layer, learning more abstract to more complex features is aimed at Chollet mentioned before [54]. Max Pooling is also used in these layers for down-streaming data and maintaining dependencies. As mentioned in DanHAR [20] before, also fine-tuning channel-wise attention is aimed too. Also, Dropout Algorithm applied in each convolution block for generating randomization, thus making a more general method is aimed.

2.4 ATTENTION LAYER

At the attention layer using the Multi-Head Attention mechanism, maintaining long-term dependencies is aimed. Instead of LSTM or RNN, attention mechanisms performed faster too.

2.5 FEED FORWARD LAYER

A traditional Feed Forward Neural Network ended with Softmax classifier is used for meaning output of sequences. The Dropout process is used in hidden layers for overcoming the overfitting problem.

CHAPTER III

EXPERIMENTAL SETUP

3.1 DATASET

Açıcı et al. [58] drew attention to the widespread use of wearable devices and the advantages of identifying the different contexts of people who use them. They collected motion data from wrist-worn sensors and classified these data for different contexts such as activity recognition and person recognition

Activities are collected with a 52Hz sample rate, and there are 28 attributes for each sample as raw and calibrated. Timestamp (milliseconds), low noise and wide range accelerometer samples for x, y and z axes(m/s^2), gyroscope samples for x, y and z axes ($°/s^2$), magnetometer samples for x, y and z axes ($°/s^2$) and voltage sample (mV) types are contained with their raw data.

Their dataset contains time series of wrist-worn accelerometer, gyroscope and magnetometer sensor data collected from 30 different people(aysu, bahadir, baris, berrak, denizhan, didem, dilara, doguhan, ezgi, furkan, hande, hasan, hatice, kenan, mert, mertdem, merve, nazli, nihal, nusret, onder, onur, onurbes, ozkan, sahika, salih, selcuk, sena, seren, tolga) whose names replaced with fake ones perform nine different activities listed below.

- **Chopping**: Chopping thick Doug with a small knife for about 60 seconds.
- **Cleaning Table**: Wiping the 150 cm×110 cm table with a small cloth for 30 seconds, starting from the upper left corner two times without holding hands.
- **Cleaning Window**: Wiping the 140 cm×55 cm table with a small cloth for 30 seconds, starting from the upper left corner two times without holding hands.
- **Drinking Water**: Drinking all water from a 33 cc porcelain glass in approximately
  30 seconds.
- **Eating Soup:** Eating soup with a spoon for about 50 seconds from a 33 cc porcelain cup. For the measurements, water was used instead of soup.

- **Kneading Dough:** Kneading about 30g of dough for 60 seconds.
- **Using a Tablet Computer:** Playing the first level of inflating balloons game on the tablet in 60 seconds.
- **Using a Computer Mouse:** Moving 20 files in a folder on the left of the computer screen to another folder on the right of the screen within 60 seconds by dragging and dropping using the mouse.
- **Writing with a Pen:** Writing 359 characters of text on A4 paper in 100 to 120 seconds using a standard pen.
- **Writing with a Keyboard:** Writing 359 characters of text with the keyboard in 80 to 100 seconds.

These listed activities have a satisfying number of labels for each activity sample, as shown in Figure 3.1.

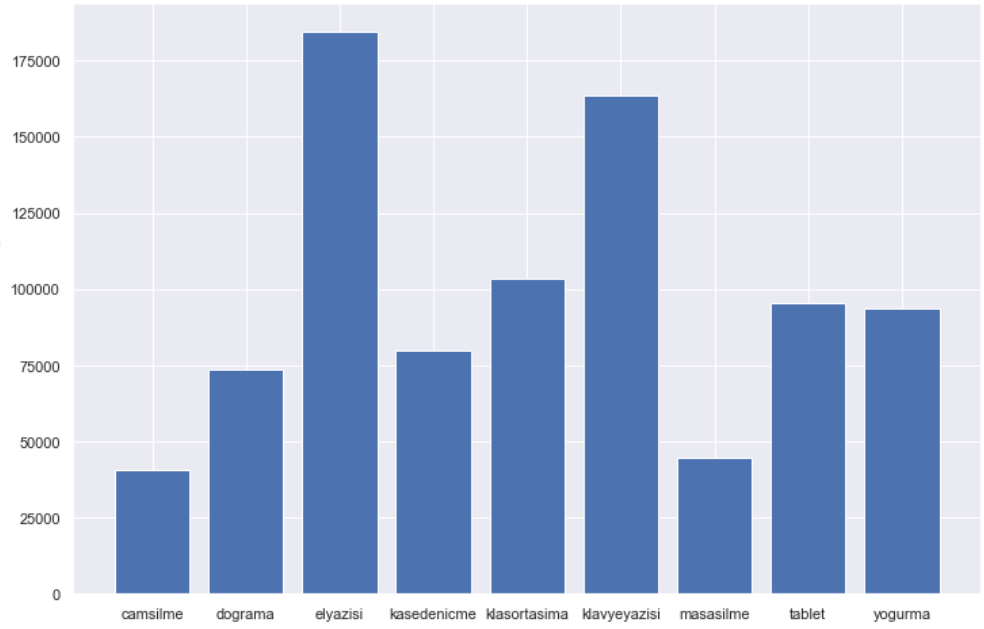


Figure 3.1: Number of Labels for Activities of HANDY Dataset

Furthermore, labels of different persons for different activities are given the heatmap represented in Figure 3.2.

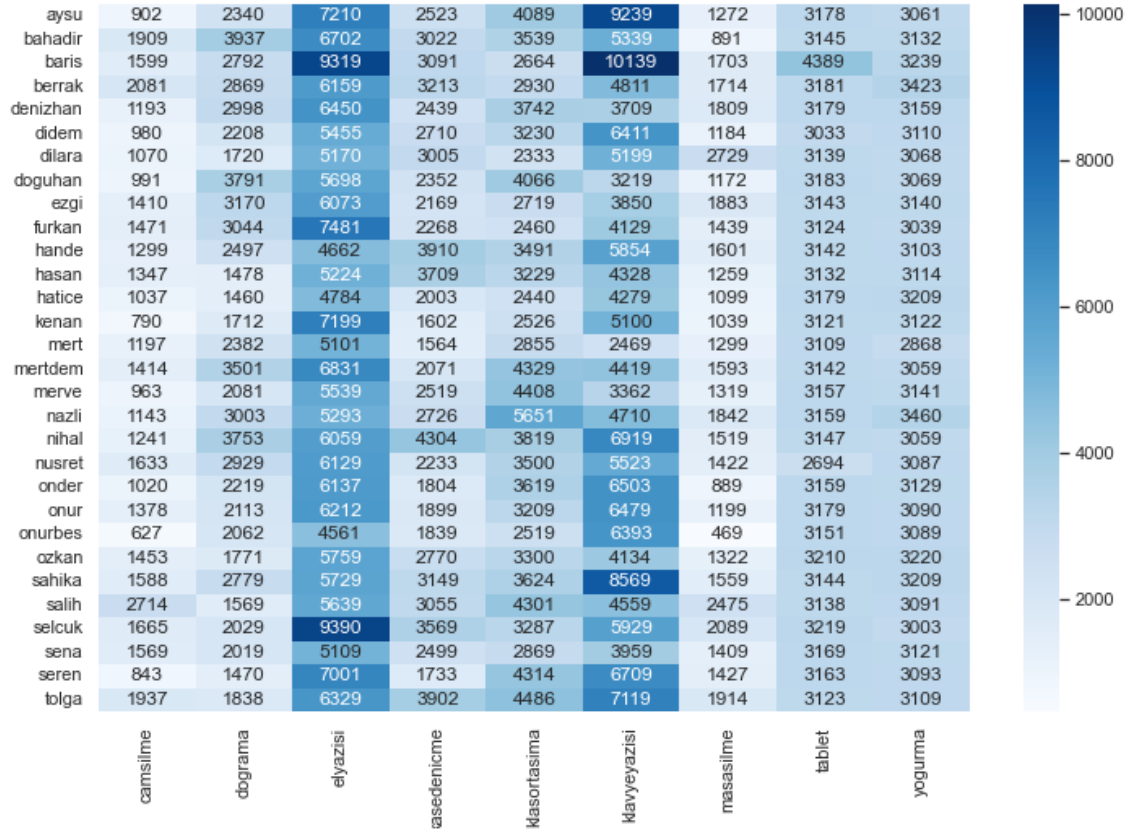


Figure 3.2: Number of Labels for Person vs. Activity in HANDY Dataset

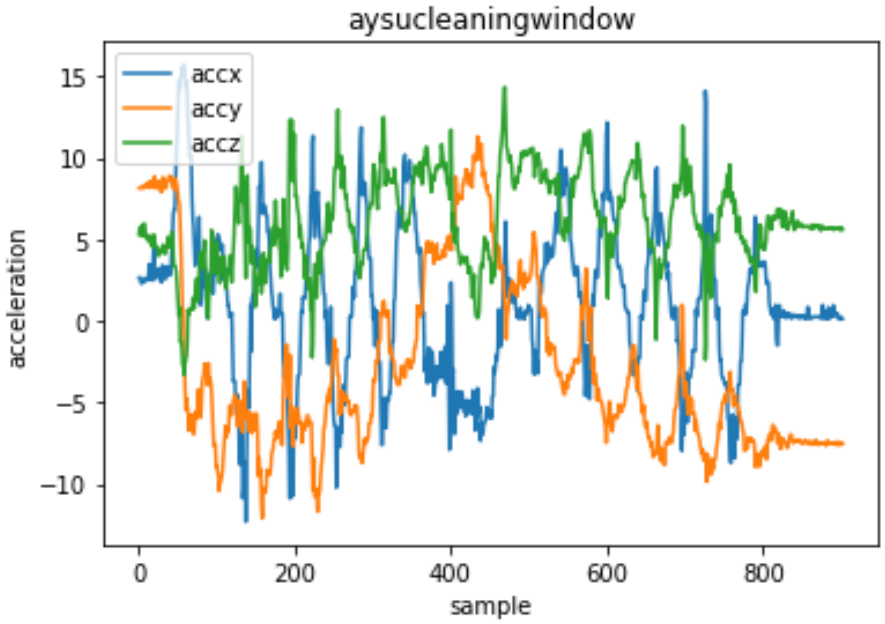


Figure 3.3: Accelerometer Data of Cleaning Window Activity by Aysu

3.1.1 Dataset Usage

In this thesis, only low noise accelerometer data values are used for classification. Sample accelerometer measurement data is given below in Figure 3.3, which is cleaning activity performed by *aysu*, where *accx* is accelerometer data of x-axis and other lines representing corresponding axes values.

3.2 HARDWARE SPECIFICATIONS

With the power of Tensorflow, the proposed model was developed with Python language in Windows Operating System. The development media have 16 GB Ram and 2.60GHz i7-6700HQ CPU. Although GPU support of Tensorflow, the model performance in less capable hardware configuration is examined.

3.3 EXPERIMENTS

There are two different experiment groups performed. One is for activity classification, and the other is for person classification concerning specific activities. For all experiments offered, the model was tested with the following modifications:

- Using LSTM layer instead of Attention Layer,
- Using file levels Convolution Layer instead of three,
- Without Optimization in Feed Forward network.

For the first experiment group, all accelerometer data rows are processed. These rows are labeled concerning activity and the person who performed the related activity, and then two label sets are generated for each row. Only related activity data is processed and labeled with persons who performed the related activity for the second experiment group.

There are wrong formatted data cells in the accelerometer columns in the HANDY dataset. During the data preprocessing, the rows containing incorrectly formatted data are ignored. Also, test and train data split in the preprocessing phase. 75% of the time series at the beginning is used for generating training data, and the other 25% is used to generate test data. This process is used for each person-activity pair for the first experimental group and each name for the second experiment group.

Because investigating the four-second length data, 75%-25% splitting means that 3 seconds continuous data is used for training, and the following one-second length continuous data is used to test the model.

This thesis assumed one-second length accelerometer data is enough to classify with this approach. So test data is generated by collecting one-second length windows with one data point shifted, and about 1/52Hz = 0.019 second shifted one-second length data frames are generated.

Performed experiments are designed to investigate fastly trainable networks for this purpose; 16 epochs are performed for the fitting model with the data. Moreover, the main objective of the models is classifying activities and persons who performed specific activities. For this cause, a categorical cross-entropy loss performance parameter is used to measure the proposed model's success. Finally, the computational efficiency, performance for noisy data benefits, adam optimization algorithm is used at the end of the model.

CHAPTER IV

RESULTS

4.1 EXPERIMENTAL RESULTS

All experiments are performed with modified models in addition to the proposed model. The modified model replaced Attention Layer with LSTM Layer annotated the tables with 'Lstm Replaced'. The model has a five-level Convolution Layer annotated with '5 Level Convolution' and without Adam Optimizer modifications of the reference, model annotated with 'No Optimization' in the following tables.

For the first experiment group, activity classification was performed with all persons and activities for the proposed model and its modifications. At the following Table 4.1, accuracy results and training times are represented.

Table 4.1: Results of First Experiment Group

Results of First Experiment Group				
		Elapsed Time (s)	**Validation Loss**	**Validation Accuracy**
Activity	Reference Model	808.176	0.359	0.935
	Lstm Replaced	917.023	0.33	0.954
	5 Level Convolution	791.041	0.3	0.924
	No Optimization	832.544	0.883	0.948
Person	Reference Model	810.578	1.964	0.485
	Lstm Replaced	918.615	0.654	0.857
	5 Level Convolution	788.742	1.006	0.741
	No Optimization	825.85	3.685	0.11

As shown in Table 4.1, for activity classification, regardless of who performed it, the Lstm Replaced model has the best performance. Moreover, the 5 Level Convolution model has less elapsed time. However, the Reference model has satisfying validation accuracy compared with the Lstm Replaced model and better Elapsed Time result.

At the same time, instead of higher elapsed time when compared to 5 Level Convolution modification, the proposed model has better Validation Accuracy. At last, the 'No Optimization' model has slightly better Validation Accuracy but has worse training time performance and higher Validation Loss. The proposed model without any modification has optimal performance when considering these implications.

Figure 4.1: Reference Model Performace for Activity Classification

The training history of the proposed model is given in Figure 4.1. The figure shows that the loss increases suddenly at the eleventh epoch besides spikes on loss curves, which slightly decreases the accuracy at the same epoch. However, the training process maintained its accuracy in the following epochs.

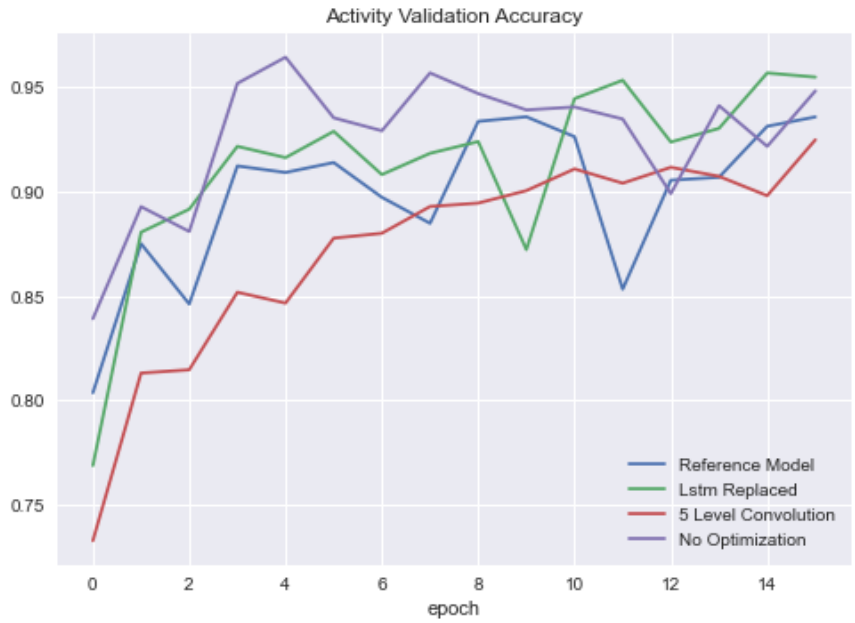


Figure 4.2: Validation Accurices of Proposed and Modified Models

As shown in Figure 4.2, 5 Level Convolution modification has more smooth accuracy performance than others; however, other models have better validation accuracy rates at the end of the training. Nevertheless, as shown in Figure 4.3, the 'No Optimizer' version of the proposed model has unexpected loss function spikes. This nature can be repeated for any epoch when using more epochs. The risks of the 'No Optimizer' version and training time of the 'Lstm Replaced' version of the proposed model are not ideal compared with the reference model.

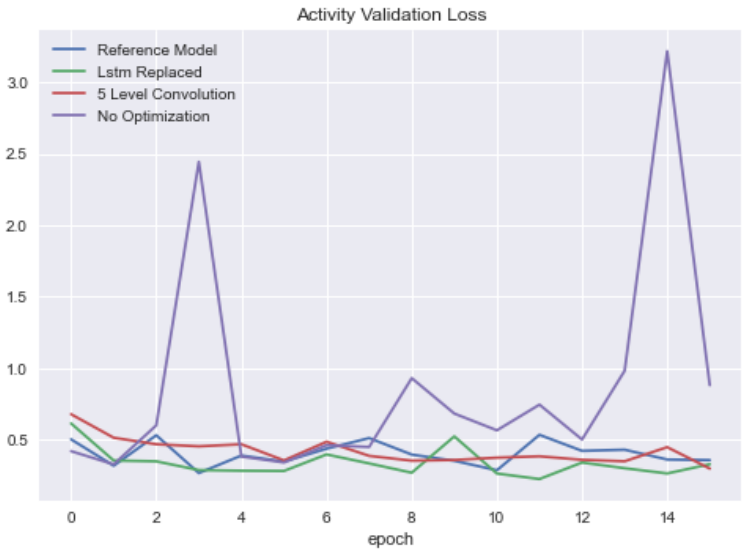


Figure 4.3: Validation Accurices of Proposed and Modified Models

Figure 4.4: Reference Model Performance for Person Classification

The proposed model has poor performance for person classification without splitting activities, as shown in Figure 4.4. Loss values are too high, and accuracy values are not satisfied. However, the 'Lstm Replaced' version model has satisfying accuracy values but high validation loss values as shown in Figure 4.5. Nevertheless, all four models have worse validation losses; as a result, the proposed model is not available for person classification for using different activity types together.

Figure 4.5: Lstm Replaced Model Performance for Person Classification

Activity-specific person classification results are shown in Table 2. *Tablet* refers to 'Using a Tablet Computer' activity, *Pen* refers to 'Writing with a Pen' activity, and *Keyboard* refers to 'Writing with a Keyboard' activity.

Table 4.2: Results of Second Experiment Group

Results of Second Experiment Group				
		Elapsed Time (s)	**Validation Loss**	**Validation Accuracy**
Tablet	Reference Model	90.003	0.011	0.995
	Lstm Replaced	101.406	0.722	0.887
	5 Level Convolution	87.919	0.262	0.966
	No Optimization	91.64	2.38	0.273
Pen	Reference Model	89.98	2.019	0.368
	Lstm Replaced	101.897	0.551	0.921
	5 Level Convolution	87.895	0.377	0.844
	No Optimization	91.596	3.128	0.1
Keyboard	Reference Model	90.165	0.421	0.903
	Lstm Replaced	101.982	0.875	0.877
	5 Level Convolution	87.914	0.378	0.908
	No Optimization	91.767	1.622	0.49

For *Tablet* experiments, the proposed model has the best Validation accuracy with satisfying training time. For Pen experiments, the surprisingly proposed model has worse Validation Accuracy. However, the Lstm Replaced version has satisfying Validation Accuracy but has bad Validation Loss. The proposed model has satisfying results for Keyboard experiments, and 5 Level Convolution modification has the best results.

When all the experiment results are examined, it can be concluded that the proposed model has the most relevant results for training time validation accuracy and validation loss parameters except Pen experiments. For a detailed explanation, model training histories can be investigated.

Figure 4.6: Person Classification Performance of the Proposed Model for Writing with a Pen Activity

The proposed model has poor person identification performance for the 'Writing with a Pen' activity, as shown in Figure 4.6, causes of high loss rates and low accuracy rates. However, the '5 Level Convolution' modification has a smooth loss curve and acceptable accuracy rate at the end of the model as shown in Figure 4.7. Nevertheless, unexpected spikes in the high loss curve whose mean value is high.

Figure 4.7: Person Classification Performance of the '5 Level Convolution' Modification for 'Writing with a Pen' Activity

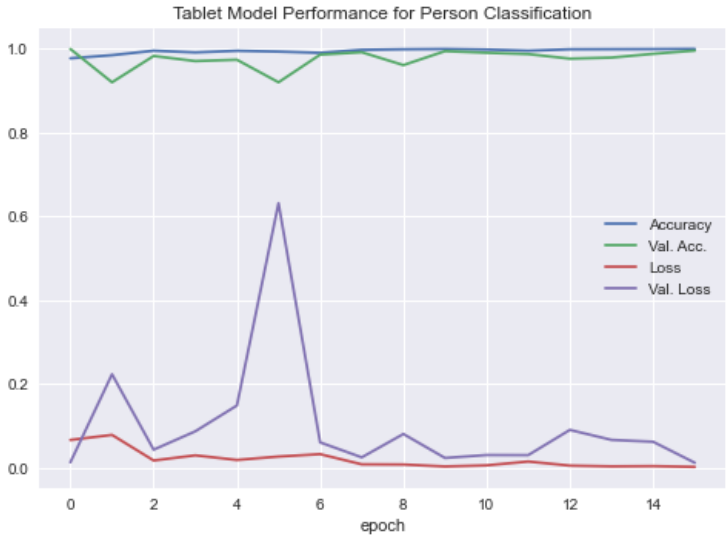


Figure 4.8: Person Classification Performance of the Proposed Model for 'Using a Tablet Computer' Activity

Person identification performance of the proposed model for 'Using a Tablet Computer' activity has enough good learning curves without validation loss value in the fifth epoch, as shown in Figure 4.8. Moreover, as shown in Figure 4.9, the proposed model has acceptable learning performance values for the ‘Writing with a Keyboard’ activity. Although the loss curve is relatively smooth, the validation loss curve has spikes.

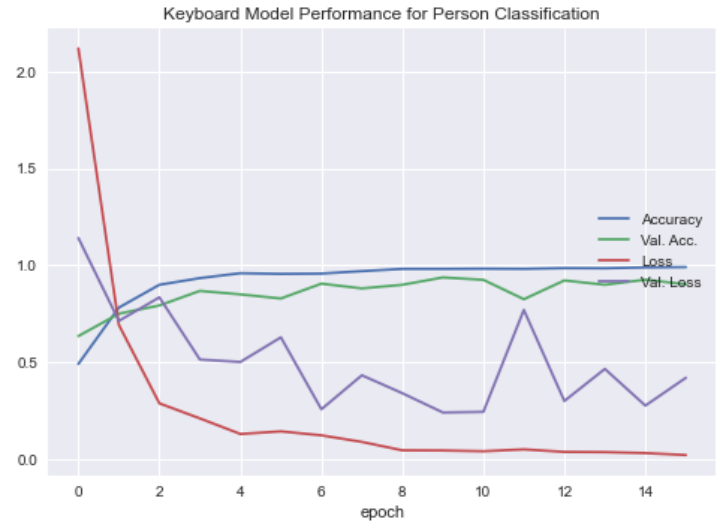


Figure 4.9: Person Classification Performance of the Proposed Model for 'Writing with a Keyboard' Activity

4.2 EMPIRICAL RESULTS

When the proposed model results are compared with the work of Açıcı et al., whose work is the first for using the HANDY dataset, the proposed model has better performance except person identification for 'Writing with a Pen' activity shown in Table 3,4,5 and 6. However, increasing the Convolution Layers from three to five, the modified version of the proposed model shows better performance against the work of Açıcı et al..

Table 4.3: Activity Classification Accuracies

Model	**Accuracy (%)**
kNN	44.7
AdaBoost	21.2
Decision Tree	54.1
Random Forest	72.2
Proposed Method	**93.5**

Table 4.4: Person Identification Accuracies for Using a Tablet Computer Activity

Model	**Accuracy (%)**
kNN	26.4
AdaBoost	6.3
Decision Tree	59.9
Random Forest	78.4
Proposed Method	**99.5**

Table 4.5: Person Identification Accuracies for Writing with a Pen Activity

Model	**Accuracy (%)**
kNN	22.1
AdaBoost	8.2
Decision Tree	82.7
Random Forest	82.9
Proposed Method	**36.8**
Modified Method (5 Level Convolution)	***84.4***

Table 4.6: Person Identification Accuracies for Writing with a Keyboard Activity

Model	**Accuracy (%)**
kNN	20.3
AdaBoost	8.9
Decision Tree	55.1
Random Forest	67.8
Proposed Method	**90.3**

CHAPTER V

CONCLUSION

5.1 ACHIEVEMENTS

As given in Results Section, the proposed model has better performance than the Machine Learning models used in the work of Açıcı et al. except for the person identification for 'Writing with a Pen Activity'. Nevertheless, the number of Convolutional Layers increased from three to five for improving the model, and the modified version of the proposed model shows better results against the work of Açıcı et al..

Using single-dimensional convolution layers for feature extraction of time series data and relating these data with Multi-Head Attention Layer with covering long-term dependencies before Feed Forward Neural Network improved activity classification and identification of person performance for the HANDY dataset.

5.2 LIMITATIONS OF MODEL

Due to the performance of the proposed model having satisfying results, as shown in validation loss graphics, there are overshoot points. During the training phase of the model, in the worst-case scenarios where the epoch number coincides with these over-shoot points, the success of the model will decrease considerably.

5.3 FUTURE WORKS

The proposed model can be modified for future works to eliminate overshoots during the training phase, and more consistent models can be trained. Also, as in the given works in the *Recent Works* section, input data can be concentrated to the output of the attention layer before Feed Forward Neural Network can improve the model performance, especially for long-term dependencies

REFERENCES

[1] J. Yu *et al.*, "A Discriminative Deep Model with Feature Fusion and Temporal Attention for Human Action Recognition," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2977856.

[2] M. Dzieżyc, M. Gjoreski, P. Kazienko, S. Saganowski, and M. Gams, "Can we ditch feature engineering? End-to-end deep learning for affect recognition from physiological sensor data," *Sensors (Switzerland)*, vol. 20, no. 22, 2020, doi: 10.3390/s20226535.

[3] G. Lai, X. Lou, and W. Ye, "Radar-Based Human Activity Recognition With 1-D Dense Attention Network," *IEEE Geoscience and Remote Sensing Letters*, 2021, doi: 10.1109/LGRS.2020.3045176.

[4] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey," *IEEE Access*, vol. 8. 2020. doi: 10.1109/ACCESS.2020.2986329.

[5] M. Abbas, M. Saleh, and R. le Bouquin Jeannès, "A Hybrid Solution for Human Activity Recognition: Application to Wrist-Worn Accelerometry," in *International Conference on Advances in Biomedical Engineering, ICABME*, 2019, vol. 2019-October. doi: 10.1109/ICABME47164.2019.8940233.

[6] P. M. Scholl and K. van Laerhoven, "A feasibility study of wrist-worn accelerometer based detection of smoking habits," 2012. doi: 10.1109/IMIS.2012.96.

[7] M. Nguyen, L. Fan, and C. Shahabi, "Activity Recognition Using Wrist-Worn Sensors for Human Performance Evaluation," 2016. doi: 10.1109/ICDMW.2015.199.

[8] S. Mehrang, J. Pietilä, and I. Korhonen, "An activity recognition framework deploying the random forest classifier and a single optical heart rate monitoring and triaxial accelerometer wristband," *Sensors (Switzerland)*, vol. 18, no. 2, 2018, doi: 10.3390/s18020613.

[9] M. S. Afzali Arani, D. E. Costa, and E. Shihab, "Article human activity recognition:

A comparative study to assess the contribution level of accelerometer, ecg, and ppg signals," *Sensors*, vol. 21, no. 21, 2021, doi: 10.3390/s21216997.

[10] N. Hegde, M. Bries, T. Swibas, E. Melanson, and E. Sazonov, "Automatic Recognition of Activities of Daily Living Utilizing Insole-Based and Wrist-Worn Wearable Sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, 2018, doi: 10.1109/JBHI.2017.2734803.

[11] S. Konak, F. Turan, M. Shoaib, and O. D. Incel, "Feature engineering for activity recognition from wrist-worn motion sensors," 2016. doi: 10.5220/0006007100760084.

[12] Z. Chen, S. Xiang, J. Ding, and X. Li, "Smartphone sensor-based human activity recognition using feature fusion and maximum full a posteriori," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, 2020, doi: 10.1109/TIM.2019.2945467.

[13] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao, "A federated learning system with enhanced feature extraction for human activity recognition," *Knowledge-Based Systems*, vol. 229, 2021, doi: 10.1016/j.knosys.2021.107338.

[14] H. Zhang, Z. Xiao, J. Wang, F. Li, and E. Szczerbicki, "A Novel IoT-Perceptive Human Activity Recognition (HAR) Approach Using Multihead Convolutional Attention," *IEEE Internet of Things Journal*, vol. 7, no. 2, 2020, doi: 10.1109/JIOT.2019.2949715.

[15] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A Semisupervised Recurrent Convolutional Attention Model for Human Activity Recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, 2020, doi: 10.1109/TNNLS.2019.2927224.

[16] Z. N. Khan and J. Ahmad, "Attention induced multi-head convolutional neural network for human activity recognition," *Applied Soft Computing*, vol. 110, 2021, doi: 10.1016/j.asoc.2021.107671.

[17] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors Journal*, vol. 19, no. 17, 2019, doi: 10.1109/JSEN.2019.2917225.

[18] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "Attnsense: Multi-level attention mechanism for multimodal human activity recognition," in *IJCAI International Joint Conference on Artificial Intelligence*, 2019, vol. 2019-August. doi:

10.24963/ijcai.2019/431.

[19] K. Chandrasekaran, W. Gerych, L. Buquicchio, A. Alajaji, E. Agu, and E. Rundensteiner, "CARTMAN: Complex Activity Recognition Using Topic Models for Feature Generation from Wearable Sensor Data," 2021. doi: 10.1109/SMARTCOMP52413.2021.00026.

[20] W. Gao, L. Zhang, Q. Teng, J. He, and H. Wu, "DanHAR: Dual Attention Network for multimodal human activity recognition using wearable sensors," *Applied Soft Computing*, vol. 111, 2021, doi: 10.1016/j.asoc.2021.107728.

[21] S. Mekruksavanich and A. Jitpattanakul, "Deep convolutional neural network with rnns for complex activity recognition using wrist-worn wearable sensor data," *Electronics (Switzerland)*, vol. 10, no. 14, 2021, doi: 10.3390/electronics10141685.

[22] H. Heydarian, P. v. Rouast, M. T. P. Adam, T. Burrows, C. E. Collins, and M. E. Rollo, "Deep learning for intake gesture detection from wrist-worn inertial sensors: The effects of data preprocessing, sensor modalities, and sensor positions," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3022042.

[23] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "DeepSense: A unified deep learning framework for time-series mobile sensing data processing," 2017. doi: 10.1145/3038912.3052577.

[24] I. Kiprijanovska, H. Gjoreski, and M. Gams, "Detection of gait abnormalities for fall risk assessment using wrist-worn inertial sensors and deep learning," *Sensors (Switzerland)*, vol. 20, no. 18, 2020, doi: 10.3390/s20185373.

[25] W. Zhang, T. Zhu, C. Yang, J. Xiao, and H. Ning, "Sensors-based Human Activity Recognition with Convolutional Neural Network and Attention Mechanism," in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2020, vol. 2020-October. doi: 10.1109/ICSESS49938.2020.9237720.

[26] S. Wang and X. Zhu, "A hybrid deep neural networks for sensor-based human activity recognition," 2020. doi: 10.1109/ICACI49185.2020.9177818.

[27] J. He, Q. Zhang, L. Wang, and L. Pei, "Weakly Supervised Human Activity Recognition from Wearable Sensors by Recurrent Attention Learning," *IEEE Sensors Journal*, vol. 19, no. 6, 2019, doi: 10.1109/JSEN.2018.2885796.

[28] M. Zeng *et al.*, "Convolutional Neural Networks for human activity recognition

using mobile sensors," 2015. doi: 10.4108/icst.mobicase.2014.257786.

[29] S. Chernbumroong, A. S. Atkins, and H. Yu, "Activity classification using a single wrist-worn accelerometer," 2011. doi: 10.1109/SKIMA.2011.6089975.

[30] D. Buffelli and F. Vandin, "Attention-based deep learning framework for human activity recognition with user adaptation," *IEEE Sensors Journal*, vol. 21, no. 12, 2021, doi: 10.1109/JSEN.2021.3067690.

[31] B. Sun, M. Liu, R. Zheng, and S. Zhang, "Attention-based LSTM Network for wearable human activity recognition," in *Chinese Control Conference, CCC*, 2019, vol. 2019-July. doi: 10.23919/ChiCC.2019.8865360.

[32] C. Betancourt, W. H. Chen, and C. W. Kuan, "Self-Attention Networks for Human Activity Recognition Using Wearable Devices," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2020, vol. 2020-October. doi: 10.1109/SMC42975.2020.9283381.

[33] M. Zeng *et al.*, "Understanding and improving recurrent networks for human activity recognition by continuous attention," 2018. doi: 10.1145/3267242.3267286.

[34] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-December.

[35] K. Kongsil, J. Suksawatchon, and U. Suksawatchon, "Physical Activity Recognition Using Streaming Data from Wrist-worn Sensors," 2019. doi: 10.1109/INCIT.2019.8912130.

[36] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal, "Design considerations for the WISDM smart phone-based sensor mining architecture," 2011. doi: 10.1145/2003653.2003656.

[37] M. Panwar *et al.*, "CNN based approach for activity recognition using a wrist-worn accelerometer," 2017. doi: 10.1109/EMBC.2017.8037349.

[38] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Complex human activity recognition using smartphone and wrist-worn motion sensors," *Sensors (Switzerland)*, vol. 16, no. 4, 2016, doi: 10.3390/s16040426.

[39] V. Nunavath *et al.*, "Deep learning for classifying physical activities from accelerometer data," *Sensors*, vol. 21, no. 16, 2021, doi: 10.3390/s21165564.

[40] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," 2013.

[41] D. Tarasevicius and A. Serackis, “Deep Learning Model for Sensor based Swimming Style Recognition,” 2020. doi: 10.1109/eStream50540.2020.9108849.

[42] A. Reiss and D. Stricker, “Introducing a new benchmarked dataset for activity monitoring,” 2012. doi: 10.1109/ISWC.2012.13.

[43] P. Zappi *et al.*, “Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008, vol. 4913 LNCS. doi: 10.1007/978-3-540-77690-1_2.

[44] A. Stisen *et al.*, “Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition,” 2015. doi: 10.1145/2809695.2809718.

[45] O. Banos *et al.*, “mHealthDroid: A novel framework for agile development of mobile health applications,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8868, 2014, doi: 10.1007/978-3-319-13105-4_14.

[46] F. J. Ordóñez and D. Roggen, “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,” *Sensors (Switzerland)*, vol. 16, no. 1, 2016, doi: 10.3390/s16010115.

[47] R. Chavarriaga *et al.*, “The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition,” *Pattern Recognition Letters*, vol. 34, no. 15, 2013, doi: 10.1016/j.patrec.2012.12.014.

[48] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, 1997, doi: 10.1109/78.650093.

[49] M. Zhang and A. A. Sawchuk, “USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors,” 2012.

[50] K. Xu *et al.*, “Show, attend and tell: Neural image caption generation with visual attention,” in *32nd International Conference on Machine Learning, ICML 2015*, 2015, vol. 3.

[51] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” 2018. doi: 10.1109/CVPR.2018.00745.

[52] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *Journal of*

Machine Learning Research, vol. 3, no. 4–5, 2003, doi: 10.1016/b978-0-12-411519-4.00006-9.

[53] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," 2008. doi: 10.1145/1409635.1409638.

[54] François Chollet, *Deep Learning with Python*. Manning Publications Co, 2018.

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[56] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, 1997, doi: 10.1162/neco.1997.9.8.1735.

[57] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015.

[58] K. Açıcı, Ç. B. Erdaş, T. Aşuroğlu, and H. Oğul, "HANDY: A benchmark dataset for context-awareness via wrist-worn motion sensors," *Data*, vol. 3, no. 3, 2018, doi: 10.3390/data3030024.

[59] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, 2005, doi: 10.1109/TPAMI.2005.159.