1

A Sequential-based Analytical Approach for Nurse Care Activity Forecasting

Md Mamun Sheikh* [1], Shahera Hossain [2], Md Atiqur Rahman Ahad [3]

[1] Department of Electrical and Electronic Engineering, University of Dhaka, Bangladesh

[2] Department of Computer Science Engineering, University of Asia Pacific, Bangladesh

[3] Department of Computer Science and Digital Technologies, University of East London, UK

Abstract

Activity forecasting is an exploding area of research due to its enormous applications in navigation systems, human-computer interaction, and health care services. Comprehensive care is ensured by properly classifying and forecasting caregivers' activities, promoting professional development, and enhancing service standards. The Fourth Nurse Care Activity Recognition Challenge addresses the new challenge of forecasting future activities utilizing prior sequences. For this challenge, we proposed a comprehensive, sequential-based analytical approach capable of recognizing and forecasting multi-label activities conducted by nurses. Our methodology also incorporates numerous pre-processing and post-processing strategies by leveraging the sequence of activities. We have attained excellent performance metrics with 99±0.54% accuracy, 90.17±2.36% sensitivity, and 99.62±0.17% specificity on multi-label activity forecasting. Compared to all other classifiers and baseline estimators,

[1]mamunsheikhdueee52@gmail.com
[2]shaherahossain@gmail.com
[3]atiqahad@gmail.com

Binary Relevance utilizing the Ada-Boost Classifier as a baseline estimator consistently outperforms others.

Keywords— Nurse Care Activity; Activity Recognition; Activity Forecasting; Binary Relevance; Ada-Boost

1 Introduction

Due to the dramatic evolution of technologies and sophisticated machine learning algorithms, the number of care facilities is expanding continually. Because of the need for caregiver human resources, enhancing the efficiency of care services is essential. The assessment of caregivers' behaviors is vital in enhancing service standards. It also provides trustworthy care services and minimizes hospitalization time [1, 2].

Activity classification and forecasting utilizing mobile sensors are feasible owing to their low cost, easy maintenance, and reduced processing capabilities. There are many more challenges to forecasting caregivers' behaviors; for instance, nurses have to do a vast range of tasks and there is intra-class variability across the same classes. In addition, complex activity sequences, discontinuous activities, and other complications associated with the data acquisition protocol make the task more challenging.

In this research, we have developed a complete, sequential-based technique for forecasting nurse care activities. The objective of our work is to automate and enhance care facilities. The primary contributions of our methodology are:

- A comprehensive problem formulation strategy and solution employing a sequential-based approach.
- Numerous pre-processing and post-processing techniques capable of finding optimum performance.

Activity classification utilizing embedded mobile sensors is ubiquitous owing to its advantages over conventional sensors, showcasing the results by M. Berchtold et. al. and Z. He et. al. [3, 4]. However, activity forecasting using sensor data is novel in this domain. The fourth Nurse Care Activity Recognition Challenge accommodates the challenge of predicting future activity employing prior information [5]. Many scholars in this area have published numerous accomplishments utilizing machine learning and deep learning approaches. Sozo Lab organized competitions in healthcare activity recognition to enhance the service quality of care facilities. In 2021, the winner of the competition [6] proposed a features-based approach to identify complicated nurses' actions using the Random Forest Classifier (RFC) utilizing raw accelerometer data [7]. They adjusted the hyper-parameters of their models and experimented with various window settings while segmenting the raw data. M. Ashikuzzaman Kowshik [8] offered different feature extraction approaches, a feature selection methodology, and a Light Gradient Boosting Classifier to attain outstanding results. In [9] paper, Zubair Rahman Tusar proposed several signal processing techniques and overlapping windowing approaches to identify the nurses' activities using RFC. Arafat Rahman [10] presented a cost-sensitive hybrid ensemble classifier to handle the imbalanced class issue in the nursing care domain. They also integrated several base classifier outputs using a stacked generalized technique.

For classification issues employing the RFC in the competition of 2020, Promit Basak [11] used a feature-based conventional machine learning model. They attained a reasonable accuracy of 65% using a stratified 10-fold cross-validation method. They used an RFC from another algorithmic pipeline provided by [10] for their re-sampling, feature selection, and validation, and they achieved a cross-validation accuracy of 65.9%.

In the 2019 competition, Md. Eusha Kadir [12] used a collection of K-Nearest Neighbors (KNN) classifiers on various hand-created features extracted from raw data. Xin Cao et. al. [13] provided another 3D motion capture pipeline based on the Spatial-Temporal Graph Convolutional Network (ST-GCN) to classify activities. For recognizing nurse action[14], Md. Nazmul Haqu presented a two-layer stacked Gated Recurrent Unit (GRU) module with a context attention mechanism.

When processing sensor data, deep learning-based models perform well compared to conventional machine learning models. Several deep learning models, including Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), have successfully recognized human behaviors in a recent study. Inoue et al. proposed that combining CNN and RNN architecture may perform with better precision [15]. Zia Uddin et al. [16] extracted the Gaussian kernel-based PCA features, fed them into the CNN architecture, and found promising results. With the ability to extract and categorize features, Charissa Ann Ronao et. al. [17] introduced a novel CNN architecture for multi-channel time series. They searched for a hidden pattern of human behavior across the model.

2 Dataset Description

The fourth Nurse Care Activity Recognition Challenge dataset comprises tri-axial accelerometer mobile sensors carried by caregivers. The positioning of the sensors (mobile) was at an arbitrary position, like a pocket. A total of 28 activities performed by the caregivers are further divided into the four major categories listed in Table 1.1.

Five people participated in the 2022 challenge, and the data were acquired in May and June 2018. The training and test files are split such that the identical subject exists in both the training and test files. The training files provided us with both the accelerometer data (x, y, z) and label files in different folders. Both data and label files include the timestamp but are distinct in the format as well as different time-zone settings. The caregiver file comprises the activities list and their start and finish timings. Labels and data are combined to yield the (X, y) pair.

The test file comprises several hours of data for each day and we are required to forecast the following hour's data utilizing these prior sequences. In the submission file, they offer us the list of activities that we have to forecast using the test data. Each hour prediction output may contain the multiple activities performed within that particular hour which transforms our challenge into a multi-label classification/forecasting problem.

TABLE 1.1: List of Activities Performed by Caregivers

Activity Name	**Number of Activities**	**Label**	**Activity Description**
Activity of Direct Care	18	1	Vital
		2	Meal/Medication
		3	Oral Care
		4	Excretion
		5	Bathing/Wiping
		6	Treatment
		7	Morning Gathering/Exercises
		8	Rehabilitation/Recreation
		9	Morning Care
		10	Daytime User Response
		11	Night Care
		12	Nighttime User Response
		13	Family/Guest Response
		14	Outing Response
		19	Get Up Assistance
		20	Change Dressing Assistance
		21	Washing Assistance
		27	Emergency Response Such as Accident
Activities of Residence Cleaning	4	15	Linen Exchange
		16	Cleaning
		23	Preparation And Checking of Goods
		24	Organization of Medications
Documentation / Communication Activities	4	17	Handwriting Recording
		18	Delegating/Meeting
		22	Doctor Visit Correspondence
		25	Family/Doctor Contact
Other Activities	2	26	Handwriting Recording
		28	Delegating/Meeting

3 Methodology

The methodology lies at the core of each real-world problem-solving effort. A superior technique may boost productivity in any circumstance. In this part, we discussed our procedure for accurately classifying and predicting future activities in the nursing care area utilizing prior actions. Our paradigm is structured into numerous sections, including:

- Data pre-processing
- Problem formulation
- Sequences generation and features extraction

- Model development
- Training and evaluation
- Post-processing

Using the aforementioned techniques, we may forecast the caregiver's future behaviors within a certain hour by analyzing previous and current actions. Figure 1 depicts the process diagram of our proposed methodology.

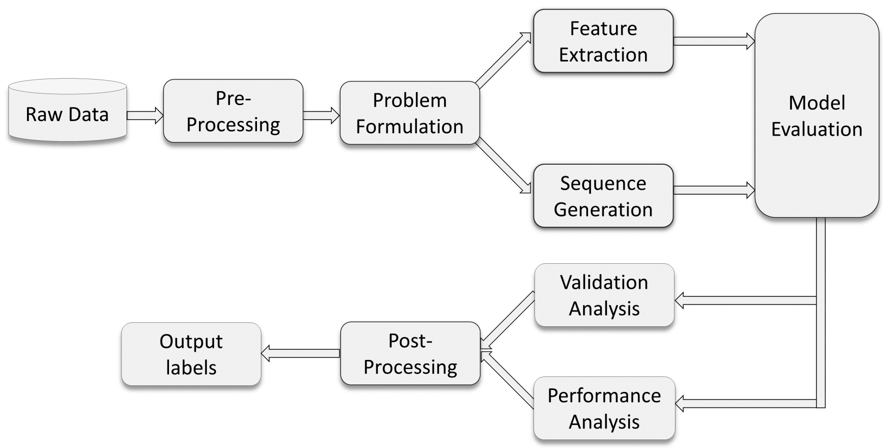

FIGURE 1: The workflow diagram of our proposed methodology

3.1 Data pre-processing

Any methodology's novelty largely depends on how we pre-process the raw data. The better the pre-processing method, the better the models' performance. The dataset we have used consists of the training, test sets, and submission files. For the training and test dataset, they provide us with tri-axial accelerometer sensor data (x, y, z) as well as activity labels. The list of activities we must predict from the test data is included in the submission files.

The main challenges in this section deal with time-series data. The timestamp they have provided in the training as well as the test set have different timezone and different formatting. We at first converted all timezone into standard timezones like UTC. After that, we converted the all timestamp into same-time formatting. The treatment of the missing values is the second major challenge (i.e., missing timestamps or accelerometer data). Discarding the missing row is the simple solution to this issue. However, doing so will result in a reduction in the amount of the training data, which is unsatisfactory for a generalized machine learning model. We have filled in the start and finish times to ensure that the missing row has the same statistics as the activity performed by the same subject. Another good estimator can be the median of the period for that activity. The raw data must be filled in by the mean value of the particular activity performed by a specific subject.

After the data has been sorted, the raw data and labels have been combined. Numerous accelerometer data points without labels are also removed. In this circumstance, having many columns with identical start and end times is redundant. Additionally, the duplicated rows yield redundant features, which might cause the model to overfit, making the duplicated rows drop-outs.

3.2 Problem formulation

Properly addressing and resolving problems in a real-world environment is the ultimate purpose of every challenge. The main objective of this challenge is to forecast future activities using knowledge from the present. Therefore, one such strategy is utilizing the accelerometer data to predict future actions. However, only one of the five users has 19 data segments that match the label files, which is insufficient to train a machine learning model.

That is why we merely made a future prediction using the label files. The start and finish times are the only information available for each activity, which is crucial for training the models. Therefore, the difficult but feasible approach is re-generating the underlying pattern using just the start and finish timings.

We are requested to forecast the activities of a certain hour in the submission file using data from previous actions. So, based on an hourly activity prediction, we may make a decision. To determine the hourly basis of activity forecasting, we have enumerated the activities performed within a certain hour. But the concern is, can the problem be generalized using the only available activity sequence? For this reason, we have generated the data and extracted a few statistical timestamps that could assist us in forecasting future action.

We first produce the period for an activity from the list of activities, together with their start and finish timestamps. Then, from the period, we retrieved several statistical features (such as mean, variance, median, and percentile). The appendix 1 contains a list of all generated timestamps and features.

In between the start and finish times, we have generated the hourly basis data (i.e., a list of the actions that occurred during that time). It produces several data samples that are used to train the model. Even if there may be no data on a given day, it is clear that the caregiver data is not continuous. The hourly basis activities will result in a large number of samples with no activity, which will cause our model to under-fit.

Therefore, we made the decision to incorporate the hours' data from the label files. We will get the n samples for the n hours of data by using this procedure. We will get a total of $(n-1)$ samples from n hours of data if we use the present activity to forecast the activity for the next hour. There is insufficient data in these $(n-1)$ samples. Because of this, we permit multiple prior sequences rather than simply one (the present sequence).

We will be able to more effectively generalize our issue and extend the sample size with the aid of the history of multiple sequences. However, how long will it take to include the prior sequences? Because the activities carried out on the previous day do not convey enough knowledge to forecast the actions on the next day, our approach is to permit the sequences to be constrained within that single day. How are the data samples (X, y pair) produced? A detailed explanation is provided below

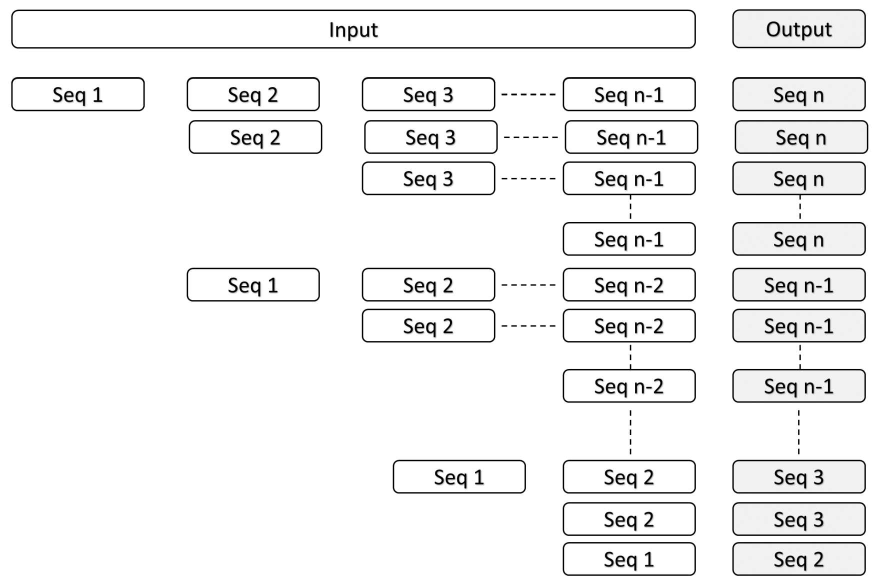

FIGURE 2: Sequence-based problem formulation approach

with illustrations.

For instance, from the 1^{st} to $(n-1)^{th}$ sequence, we may predict the n^{th} sequence (y) if we have total n sequences (n hours of data) (X). Sequences 2 through $(n-1)$ are then used to forecast the n^{th} sequence, and so on. For the following prediction, the sequences 1 through $(n-2)$ must be used, and so on. This method enables us to combine n sequences into a total of $\frac{n\cdot(n-1)}{2}$ samples. Figure 2 visually represents this technique. These sequences are then used to extract the feature and input it into the machine learning models.

3.3 Features extraction

Dealing with time-series data is quite challenging. There are two sorts of methods for resolving this problem. The first is the training using raw data, while the second is the training utilizing the extracted features. Training with raw data is challenging since we must understand the models better. The models can automatically suppress the noise, conduct data standardization, and predict the hidden patterns, which need the model to be novel and scaleable. The second one is the manual feature extraction from the raw data, enabling the model to circumvent the various processes indicated above. On the other hand, feature extraction-based models require less computation power than deep-learning models.

The hourly basis sequences enable us to generate a list of actions encompassing that particular hour and transform the problem into multi-class multi-label classification problems. We extract features from the timestamps from the hourly-based activity forecasts, which helps us generate the final sequences (X) to train the models. Several temporal statistical features are retrieved from the timestamps data, which are the time-period data. Finally, from each period, we estimated the features such an hour, minute, and second. On the other hand, we have extracted some additional features from the date-time-related timestamp, including a weekday, days-in-a-month, and week. All the specifications of the features list are provided in the Appendix A.

In addition, we have also incorporated the activity information into the features list. This activity information also helps to predict future activity or activity forecasting. As the activities are different for various users, the features' duration will also differ for different users. On the other hand, the activities list of each hour data has a different number of activities, resulting in dynamic length features for each data. The next part will present a novel way to alleviate this challenge.

3.4 Sequences Generation

Sequence classification has various real-world applications, including language translation, speech recognition, and numerous others. This problem's main challenge is dealing with the variable length input size. Similar to the language translation, each word is turned into an embedding vector similar to the feature space in our situation. Nevertheless, the feature vector has a variable input size in our instance.

We restrict the maximum number of segments to extract the features to handle the variable length input size. If an hour contains segments greater than the predefined MAX_SEG, the remaining segments are disregarded for feature extraction. On the contrary, if hourly data includes a segment more diminutive than the MAX_SEG, then the remaining segments are padded with the zero value to preserve the fixed input length.

However, we have designed our challenge to retrieve numerous historical data rather than a single-hour history (current hour). We have taken the maximum number of prior sequences to a specified value, which is often less than or equal to the MAX_SEQ value. To forecast a future value, if we have the length of the sequence less than the MAX_SEQ, the remaining sequences are padded to retain the constant input form, which is a precondition to training a machine learning model. The visual representation of the generation of sequences is depicted in Figure 3.

3.5 Model development

Divide and conquer is one of the most fundamental issue-solving strategies for dealing with a sizeable, challenging task. The problem is separated into multiple sections and recombined to obtain the appropriate objective. In our challenge formulation phase, we formulated our challenge to have many actions in an hour. This issue is a multi-class classification/forecasting problem if there is a single action each hour. However, the problem is turned into a multi-class, multi-label problem to have many activities. We have addressed this challenge by utilizing machine learning approaches. In our dataset, we have trained three distinct models: Binary-Relevance, Classifier-Chain, and Label-Power-Set. The in-depth concepts of each paradigm are given below.

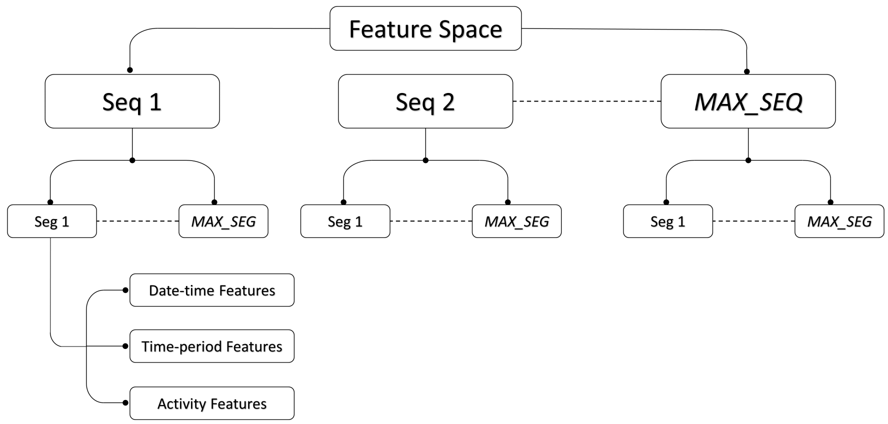

FIGURE 3: Generation of feature space utilizing sequences

3.5.1 Binary Relevance

The Binary Relevance model works based on the divide and conqueror strategy. It turns the multi-label classification issue into a binary classification problem. A multi-label classification issue is converted into L different labels. It converts the problem into L independent binary classification problems. It utilized the same underlying classifier for all classification tasks. After that, all the forecasts are combined to get the final output. It takes the union of all classifier outputs.

3.5.2 Classifier Chain

The classifier chain model employs the chaining technique to generate the final result. For improved performance, the multiple classifiers are connected into a chain. The very first classifier is trained using the training data. The next classifier is generated using the combined input and the preceding classifier. In this sequential procedure, one classifier output is utilized as an input for the next classifier in the classifier chain. This model can preserve the correlation between labels.

3.5.3 Label Power Set

The Label Power Set concept translates multi-label classification into a multi-class problem. It regenerates the labels using a unique combination of labels retrieved from the dataset. The principal purpose of this model is to assign a new value to each unique label combination and fit the newly formed labels for categorization. Thus, the classifier has allocated the new value for categorization. This strategy tends to offer us more reliability.

3.6 Post-processing

Post-processing techniques are necessary to improve the performance and robustness of any methodology. In general, both the computing cost and model complexity is improved. We used two post-processing techniques: feature selection and model ensemble. In this part, we thoroughly study how these methods assist us in achieving promising results.

3.6.1 Features selection

The feature selection approach is quite efficient with large-scale features in the dataset. It assists us in identifying the crucial features within the feature space. This method lowers both the over-fitting issue and the cost of computing. In our instance, we chose the feature importance method for feature selection.

The Random Forest Classifier (RFC) uses a tree-based strategy to find the most important features. It naturally learns how to improve the purity of the node by decreasing the Gini Impurity over the nodes. The greatest impurity drop occurs at the beginning of the nodes, whilst the slightest impurity decrease occurs at the end. We obtain the subset of the most important features by pruning the trees below a particular node. The importance of features is shown in the result section.

3.6.2 Model ensemble

An ensemble of models is a robust machine-learning technique that combines the output from different models to generate the final prediction. The ensemble members who contribute to the ensemble process are the same in our scenario. The ensemble process is done using statistical or more sophisticated methods that teach how much to trust each ensemble member.

Many methods, like averaging, max voting, and bagging, ensemble the members. We have used the max voting technique to generate the final prediction. In the problem statement, we have formulated that if there is total n hour data and we have to predict n^{th} hour activities, the prediction can be made using the total $(n-1)$ samples. Each $(n-1)$ samples can predict the n^{th} sequence. After that, the predictions from these $(n-1)$ ensemble members are ensembled to generate the final prediction using the majority voting technique. The visual illustration of how the ensemble method is performed is shown in Figure 4.

4 Results and Analysis

Analytical approaches aid in the identification of the problem and its solution. This section discusses the general performance of the various models, the rationale for their use, and numerous performance metrics.

Dealing with missing data is crucial in the pre-processing processes, as it enhances the overall performance of the models. Rather than discarding the missing values, we filled them with statistics such as the mean value. As a result, it increases the number of data samples and classification accuracy as well. The baseline model dramatically increases classification accuracy by exploiting missing variables (Binary Relevance using RFC as the base estimator without hyper-parameter tuning).

Due to insufficient samples of the raw accelerometer data, we have just utilized the

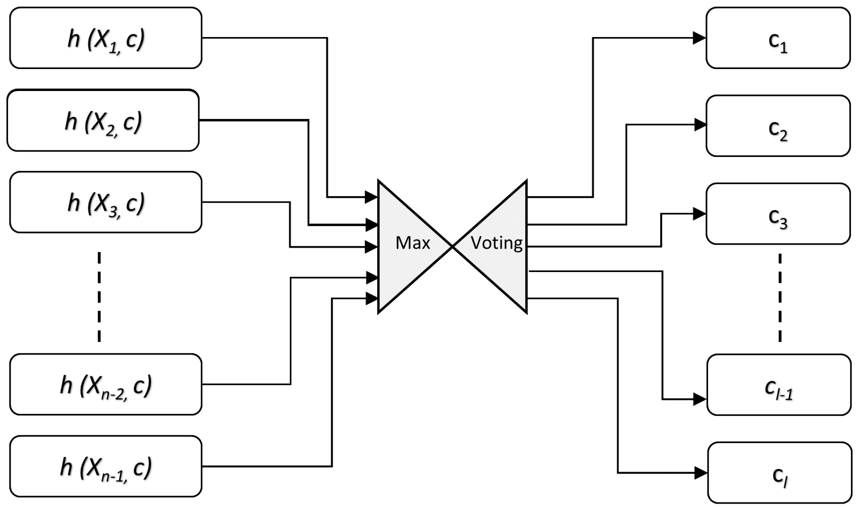

FIGURE 4: Ensemble of different output to generate final prediction

care record data from the training and test files. At first, we trained our baseline model on [5] dataset as well as [6] dataset. We have obtained significant accuracy from both datasets. To test the robustness of our models, we have used different validation methods, as discussed in the methodology section. For the first time, we used a 20% random validation split to validate the models. The baseline model's accuracy (random split) for both datasets (including test data of [5]) are shown in Tables 1.2 and 1.3, respectively. We trained the model separately for each user since the submission files had a varying amount of actions for each user.

TABLE 1.2: Accuracy and Hamming Loss for Baseline Model using 20% Random Validation Split on Dataset [5]

Dataset	**Schema**	**Accuracy (%)**					**Hamming Loss**				
		s8	s13	s14	s15	s25	s8	s13	s14	s15	s25
Dataset [5]	train	81.51	84	82.91	92.30	82.2	0.184	0.201	0.164	0.077	0.178
	test	82.32	79.89	83.51	87.43	81.50	0.176	0.164	0.164	0.125	0.185

TABLE 1.3: Accuracy and Hamming Loss for Baseline Model using 20% Random Validation Split on Dataset [6]

Dataset	**Schema**	**Accuracy (%)**					**Hamming Loss**				
		s3	s6	s12	s19	s22	s3	s6	s12	s19	s22
Dataset [6]	Train	92.31	94.15	85.31	90.63	83.25	0.076	0.058	0.146	0.093	0.167

On both datasets, our baseline model performed well. Nevertheless, the question arises, "Can this model actually handle our classification tasks?".When we just changed our validation schema from random to date-wise (using 80% of the data for training and the remaining 20% for validation), the validation accuracy of baseline models deteriorated drastically. The date-wise validation accuracy for both datasets is provided in Table 1.4 and 1.5.

TABLE 1.4: Accuracy and Hamming Loss for Baseline Model using 20% Date-wise Validation Split on Dataset [5]

Dataset	**Schema**	**Accuracy (%)**					**Hamming Loss**				
		s8	s13	s14	s15	s25	s8	s13	s14	s15	s25
Dataset [5]	Train	7.23	11.11	32.41	24.24	8.25	0.927	0.888	0.675	0.757	0.917
	Test	4.20	10.40	34.17	42.3	10.42	0.958	0.896	0.658	0.577	0.895

TABLE 1.5: Accuracy and Hamming Loss for Baseline Model using 20% Date-wise Validation Split on Dataset [6]

Dataset	**Schema**	**Accuracy (%)**					**Hamming Loss**				
		s3	s6	s12	s19	s22	s3	s6	s12	s19	s22
Dataset [6]	Train	15.15	4.20	1.11	20.05	10.45	0.848	0.958	0.988	0.799	0.895

However, how can we argue if our model is generalized or robust? The basic answer is yes, and our model is generalized and robust, as demonstrated in Table 1.2 and 1.3. However, the low accuracy of the baseline model displayed in Table 1.4 and 1.5 is due to a modest correlation between the activities performed on one day and those conducted on another. The models (or even humans) will not be able to anticipate future actions in subsequent days by using the activities being conducted now.

Therefore, it is evident that the activities predicted for a time coincide with the activities carried out by the users on that day. Because of this, we solely used the test data to create the submission files. The test data, however, only contains a small number of samples, making it impossible to train the models. To address this issue, we developed a unique sequence-generating approach covered in the problem formulation section. This method allows us to create the $\frac{n \cdot (n-1)}{2}$ from n sequences. After that, we trained the various multi-label classifiers mentioned in the methods section. We also adjusted the hyper-parameter for the RFC and Adaptive Boost Classifier baseline estimators. There is a list of hyper-parameters in Appendix 1. The models are first trained using a 20% random validation split to identify the effectiveness before being trained using the whole dataset. In Table 1.6, the validation accuracy of several models for various subjects are shown.

Additionally, we evaluated the various performance metrics given in Appendix 1. Binary Relevance outperforms the other two models. The Binary Relevance model with an Ada-Boost Classifier outperforms the two baseline estimators. Table 1.7 illustrates several performance metrics for the Binary Relevance Classifier utilizing Ada-Boost as the baseline estimator.

We used our post-processing technique in the last stage to yield the final predictions. The total $(n-1)$ sub-sequences mentioned in the problem formulation

TABLE 1.6: Performance Metrics of Different Classifiers without Hyper-parameter Tuning using Ada-Boost as the Baseline Estimator

Dataset	Schema	Model	Estimator	Sub	Performance Metrics	
					Accuracy (%)	Hamming Loss
Dataset [5]	Test: Validation	Binary Relevance	Random Forest	s8	67.04	0.329
				s13	72.72	0.272
				s14	69.01	0.309
				s15	78.94	0.210
				s25	35.89	0.641
			Ada-Boost	s8	80.68	0.192
				s13	88.63	0.113
				s14	85.91	0.140
				s15	88.42	0.115
				s25	58.97	0.410
		Classifier Chain	Random Forest	s8	67.04	0.329
				s13	75.00	0.250
				s14	66.19	0.338
				s15	77.89	0.221
				s25	33.33	0.670
			Ada-Boost	s8	84.09	0.157
				s13	88.63	0.113
				s14	8169	0.183
				s15	87.36	0.126
				s25	53.84	0.461
		Label Power Set	Random Forest	s8	81.82	0.182
				s13	84.09	0.159
				s14	80.28	0.197
				s15	81.05	0.189
				s25	53.85	0.462
			Ada-Boost	s8	15.91	0.841
				s13	36.36	0.636
				s14	14.08	0.859
				s15	54.74	0.453
				s25	20.51	0.795

TABLE 1.7: Performance Metrics of Binary Relevance Classifier using Ada-Boost as the Baseline Estimator after Tuning Hyper-parameters

Dataset	Schema	Sub	Performance Metrics					
			Accuracy	Sensitivity	Specificity	Precision	F1-Score	Hamming Loss
Dataset [5]	Test: Valid	s8	99.91	90.93	99.76	97.65	94.17	0.001
		s13	98.88	86.84	99.72	95.65	91.03	0.011
		s14	98.20	90.86	99.28	94.84	92.81	0.017
		s15	99.05	93.75	99.71	97.56	95.61	0.009
		s25	98.99	88.47	99.65	94.07	91.18	0.010
	Test: Train	All	100.0	100.0	100.0	100.0	100.0	0.000

section can be used to forecast the n_{th} sequence. The final predictions is then created by combining the predictions from all $(n-1)$ samples. The merging is done using the majority voting technique. The prediction using a sub-sequence t is denoted as $h(X_t, c_i)$ where $t = 1, 2, ...(n-1)$ and c_i is the prediction of i^{th} class. The final predicted classes are the highest average scores calculated on all $h(X_t, c_i)$ sub-sequences predictions using the following equation:

$$c_i = argmax_i \left(\frac{1}{(n-1)} \cdot \sum_{t=1}^{n-1} h(X_t, c_i) \right) \tag{1.1}$$

The performance metrics of Binary Relevance Classifier using Ada-Boost as the baseline estimator after performing the post-processing are shown in Table 1.8.

TABLE 1.8: Performance Metrics of Binary Relevance Classifier using Ada-Boost as the Baseline Estimator using Post-processing Technique (Max Voting)

Dataset	**Schema**	**Sub**	**Performance Metrics**					
			Accuracy	Sensitivity	Specificity	Precision	F1-Score	Hamming Loss
Dataset [5]	Test: Valid	s8	100.0	100.0	100.0	100.0	100.0	0.000
		s13	99.32	89.45	99.86	96.42	92.11	0.006
		s14	98.27	90.92	99.32	94.96	93.02	0.017
		s15	99.18	94.05	99.78	96.82	96.57	0.008
		s25	99.12	88.75	99.77	94.82	92.07	0.009
	Test: Train	All	100.0	100.0	100.0	100.0	100.0	0.000

5 Conclusion

This research illustrates the capabilities of a machine learning model to predict a complicated caregiver's activities using a unique problem-solving strategy. We have acknowledged that the present activities substantially correlate with the actions conducted in the previous hours. The forecasting algorithms utilized the data on an hourly basis in our instance. However, there might be another technique other than hourly, such as a quarter of the day or a minute basis. In this context, we have used the caregiver's data solely. Nevertheless, the accelerometer data is better associated with the activities performed by the subject if we can adequately pre-process it. We have employed a sequential-based method to mine data for our machine-learning models. Another data augmentation approach, such as Generative Adversarial Network (GAN), may provide more sophisticated and correlated data, a method of choice. We merely trained the typical machine learning model to predict future actions. Other sequential-based deep learning models may handle such scenarios, such as Gated Recurrent Unit (GRU) or Long Short-Term Memory (LSTM). We have just implemented the Gini impurity-based feature selection. However, additional feature selection approaches like variance thresholds, dispersion ratio, and recursive feature elimination may greatly assist us.

An interactive process between caregivers and patients is nurse care activity forecasting. For improved healthcare services, it establishes service standards. Identifying and defining the requirements for activity forecasting is made easier with the use of nursing activity forecasting. By adjusting certain variables (such as $MAXSEQ$ and $MAXSEG$) to estimate future events, forecasting horizons for both the long-term and short-term may be established. Our proposed methodology, on the other hand, places a strong emphasis on creating healthcare activity forecasting by employing a straightforward sequential technique.

To generate the feature space, we have confined the maximum number of segments inside a specific hour to 15 and the number of sequences (history in an hour) to 10. We selected these numbers as the 75 percentile value for each situation. Any combination of MAX_SEG and MAX_SEQ may be selected to decide how our model performs. In the post-processing step, we employed ensemble techniques to give the final prediction. The ensemble is accomplished using the majority vote mechanism. However, ensemble approaches like weighted average or cost-sensitive learning are more robust.

The criteria described above may be implemented to increase the model performance. Addressing all the challenges is rather tough to integrate into our solution. In the future, we will address the issue of constructing a robust and generic activity forecasting model.

Bibliography

[1] Sozo Inoue, Paula Lago, Tahera Hossain, Tittaya Mairittha, and Nattaya Mairittha. 2019. Integrating Activity Recognition and Nursing Care Records: The System, Deployment, and a Verification Study. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 3, 3, Article 86 (September 2019), 24 pages. https://doi.org/10.1145/3351244

[2] Sheikh, Md Mamun, Faizul Rakib Sayem, and Md Atiqur Rahman Ahad. "A Residual Network with Focal Loss to Handle Class-imbalance Problem on Nurse Care Activity Recognition." In 2021 Joint 10th International Conference on Informatics, Electronics Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision & Pattern Recognition (icIVPR),Kitakyushu, Japan, pp. 1-8. IEEE, 2021, doi: 10.1109/ICIEVicIVPR52578.2021.9564165.

[3] M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke and M. Beigl, "ActiServ: Activity Recognition Service for mobile phones," International Symposium on Wearable Computers (ISWC) 2010, 2010, pp. 1-8, doi: 10.1109/ISWC.2010.5665868.

[4] Z. He and L. Jin, "Activity recognition from acceleration data based on discrete consine transform and SVM," 2009 IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 5041-5044, doi: 10.1109/ICSMC.2009.5346042.

[5] Sozo Inoue, Defry Hamdhana, Christina Garcia, Haru Kaneko, Nazmun Nahid, Tahera Hossain, Sayeda Shamma Alia, Paula Lago, May 22, 2022, "4th NURSE CARE ACTIVITY RECOGNITION CHALLENGE DATASETS", IEEE Dataport, doi: https://dx.doi.org/10.21227/vchd-s336.s

[6] Sayeda Shamma Alia, Kohei Adachi, Nhat Tan Le, Haru Kaneko, Paula Lago, Sozo Inoue, April 29, 2021, "Third Nurse Care Activity Recognition Challenge", IEEE Dataport, doi: https://dx.doi.org/10.21227/hj46-zs46.s

[7] Faizul Rakib Sayem, MD Mamun Sheikh, and Md Atiqur Rahman Ahad. 2021. Feature-based Method for Nurse Care Complex Activity Recognition from Accelerometer Sensor. In Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers (UbiComp/ISWC '21 Adjunct). Association for Computing Machinery, New York, NY, USA, 446–451. https://doi.org/10.1145/3460418.3479388

[8] M. Ashikuzzaman Kowshik, Yeasin Arafat Pritom, Md.Sohanur Rahman, Ali Akbar, and Md Atiqur Rahman Ahad. 2021. Nurse Care Activity Recognition from Accelerometer Sensor Data Using Fourier- and Wavelet-based Features. In Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers (UbiComp

'21). Association for Computing Machinery, New York, NY, USA, 434–439. https://doi.org/10.1145/3460418.3479387

[9] Zubair Rahman Tusar, Maksuda Islam, and Sadia Sharmin. 2021. Accelerometer based Complex Nurse Care Activity Recognition using Machine Learning Approach. In Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers (UbiComp '21). Association for Computing Machinery, New York, NY, USA, 452–457. https://doi.org/10.1145/3460418.3479390

[10] Arafat Rahman, Nazmun Nahid, Iqbal H assan, and M. A. R. Ahad. 2020. "Nurse care activity recognition: using random forest to handle imbalanced class problem," In Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp-ISWC '20), 419–424. DOI:https://doi.org/10.1145/3410530.3414334

[11] Promit Basak, Shahamat Mustavi Tasin, Malisha Islam Tapotee, Md. Mamun Sheikh, A. H. M. Nazmus Sakib, Sriman Bidhan Baray, and M. A. R. Ahad. 2020. "Complex nurse care activity recognition using statistical features," In Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp-ISWC '20), 384–389. DOI:https://doi.org/10.1145/3410530.3414338

[12] Md. Eusha Kadir, Pritom Saha Akash, Sadia Sharmin, Amin Ahsan Ali, and Mohammad Shoyaib. 2019. "Can a simple approach identify complex nurse care activity?," In Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct), 736–740. DOI:https://doi.org/10.1145/3341162.3344859

[13] Xin Cao, Wataru Kudo, Chihiro Ito, Masaki Shuzo, and Eisaku Maeda. 2019. "Activity recognition using ST-GCN with 3D motion data," In Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct), 689–692. DOI:https://doi.org/10.1145/3341162.3345581

[14] Md. Nazmul Haque, Mahir Mahbub, Md. Hasan Tarek, Lutfun Nahar Lota, and Amin Ahsan Ali. 2019. "Nurse care activity recognition: a GRU-based approach with attention mechanism," In Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct), 719–723. DOI:https://doi.org/10.1145/3341162.3344848

[15] Masaya Inoue, Sozo Inoue, and Takeshi Nishida. 2018. "Deep recur-rent neural network for mobile human activity recognition with highthroughput," Artificial Life and Robotics, 23, 2, 173–185.

[16] M. Z. Uddin and M. M. Hassan, "Activity Recognition for Cognitive Assistance Using Body Sensors Data and Deep Convolutional Neural Network," in IEEE Sensors Journal, 19, 19, 8413-8419, 2019, doi: 10.1109/JSEN.2018.2871203.

[17] Charissa Ann Ronao, Sung-Bae Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," Expert Systems with Applications, 59, 2016, 235-244, https://doi.org/10.1016/j.eswa.2016.04.032.

1 Appendix A

TABLE A1: List of extracted timestamps and features

Serial	Scenario	Name	Total Features
1	Timestamp	start, finish, difference, mean, standard deviation maximum, minimum, median, range, percentile, z_score	
2	Features	year, month, day, quarter, day of week, day of the year, week of the year, days in the month, is month end, is month start is the quarter end, is the quarter start, week hour, minute, seconds periods of (hour, minute, second, total seconds)	637 / sequence
3	Feature Space		637*MAX_SEQ

TABLE A2: List of hyper-parameters

Serial	Scenario	Name	Values
1	Hyper-parameters	estimators	[RFC, Ada-Boost]
		max_depth	[500,1000,2000,5000, no. of features]
		max_features	[auto,500,1000,2000,5000, no. of features]
		n_estimators	[20, 50, 100, 200, 250, 500]
2	Best parameters	estimators	Ada-Boost
		max_depth	5000
		max_features	5000
		n_estimators	200

TABLE A3: Confusion Matrix

Actual	Predicted: Positive	Predicted: Negative
Positive	TP	FN
Negative	FP	TN

Formulas for different performance metrics are given below

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{.2}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{.3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{.4}$$

$$Precision = \frac{TP}{TP + FP} \tag{.5}$$

$$F1 - score = \frac{2 \cdot Precision \cdot Sensitivity}{Precision + Sensitivity} \tag{.6}$$

$$HammingLoss = \frac{FP + FN}{TP + TN + FP + FN} \tag{.7}$$

TABLE A4: Miscellaneous Information

Name	**Description**
Dataset Used	Dataset [6, 5]
Data Used	Only Caregivers' data
Model Types	Feature-based Machine Learning Model
Classification Model	Binary Relevance with Ada-Boost Classifier
Total Features	6370+ / sample
Post-processing	Yes
Performance Metrics	Accuracy, Sensitivity, Specificity, Precision, F1-Score, Hamming Loss
Device Specification	Google Colab
Programming Languages	Python and MATLAB
Library Used	Numpy, Pandas, Matplotlib, Scipy, sklearn, skmultilearn