

Predicting Bedside Falls using Current Context

Asbjørn Danielsen

*Faculty of Engineering Science and Technology
UiT – The Arctic University of Norway*

Narvik, Norway
asbjorn.danielsen@uit.no

Bernt A. Bremdal

*Faculty of Engineering Science and Technology
UiT – The Arctic University of Norway*

Narvik, Norway
bernt.a.bremdal@uit.no

Abstract—Each year about a third of the elderly aged 65 or older experience a fall. Many of these falls could be avoided if fall risk assessment and prevention tools were available in the daily living situation. Such tools would need to use the current context as input to predict an imminent fall. This paper presents an approach predicting imminent falls using data from a roof-mounted infrared array combined with an ultrasonic sensor. The data are processed and features extracted to determine location and posture along with indicators representing movement, direction, and velocity. These features are used by a classification algorithm to create a probability matrix representing the conditional probability of an individual in the current frame being recognized in a specific location and posture. A sequence of these probability matrices are fed into four artificial intelligence constructs trained to predict the probability of a future location/posture. The resulting conditional probability is used as a fall risk indicator to predict falls. Finally, the results from the experiment are presented. The study concludes that Elman Recurrent Neural Network with adapted Teacher Forcing has very promising properties and an explanation of the findings is offered.

Keywords—*recurrent neural net, Elman net, linear ridge regression, logistic ridge regression, Bayes net, fall prediction, context aware fall prediction, ambient monitoring, current context*

I. INTRODUCTION

Each year about one third of the elderly aged 65 or older experience a fall [1], and each year older adults are hospitalized for fall-related injuries up to five times more often than other causes [2]. Between 10% and 20% of falls by the elderly result in serious injuries such as fractures or head traumas while non-fatal fall injuries are associated with considerable morbidity including decreased functioning and loss of independence [3]. While seniors in hospital or intervention settings contribute to around 20% of all registered falls, approximately 50% of all falls resulting in hospitalization occur in the home environment by community-dwelling seniors [4]. Out of these falls, around 30% occur in the bedroom [5].

A. Approaches for Prediction and Forecasting

Predicting the future/next instance(s) in a time-series of discrete values has been investigated by many. Most time-series have properties that make them statistically predictable over a period, include seasonal variations or cyclic patterns, have distinct trend variations, or even irregular fluctuations, including a degree of randomness. Forecasting is possible since

the future is dependent on the past, or analogously because there is a relationship between the future and the past. This relation is however not deterministic and can hardly be written in an analytical form. To illustrate the diversity of prediction we present a selection of such approaches below. This presentation is not exhaustive.

Statistical approaches, like ARIMA, have been used for prediction in combination with machine learning techniques. For example, Asadi et al. [6] combined data preprocessing methods, genetic algorithms and Levenberg-Marquardt algorithm in learning a back-propagation neural network.

Bayes networks and Markovian approaches model the possible states of a system, the possible transition paths between those states, and how the states are related by probabilities. As such, these approaches express the probability of a future state based on the current state and implicit the path leading to this state. None of the approaches maintains any notion of memory, but rather expresses the next state as a probability learned during training.

Feed-forward neural networks have also been used as a tool for prediction. Corani [7] used a feed forward neural network to predict air quality in Milan. He compared the results produced with the results from a pruned neural networks and lazy learning. He concluded that no significant differences could be observed.

Regression models express how a number of features correspond to some function. The relationship may be linear, logistic, polynomial, etc. and the model may be adjusted for errors by applying ridge regression techniques. Ridge regression is used when the data suffers from multi collinearity, i.e. independent variables are highly correlated. In multi collinearity, even though the least squares estimates are unbiased; their variances are large which deviates the observed value far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

Kalman filters [8] have been applied to predict the next state in linear systems. The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive and can run in

real time, which makes it very suitable in robotics motion planning and control [9] and navigation [10].

Recurrent Neural Nets (RNN) are neural networks with a closed feedback connection. The architectures range from fully interconnected nets with no distinct input layers, to partially connected nets that combine feedforward structures while some nodes provide the sequential context and receive feedback from the previous time step. The nodes providing the sequential feedback are usually referred to as context nodes. The context nodes may receive feedback from the output nodes as in Jordan RNN [11], using the hidden layer as input to the context nodes as in Elman RNN [12], or by controlling the context otherwise, either by adding layers to the network or by controlling how and when data in the context nodes are updated.

Jordan and Elman RNNs have challenges in respect to how the feedback has impact on the resulting output of the RNN and for how long. This is specifically related to learning and altering long-range dependencies and is known as the vanishing and exploding gradients. This occurs when backpropagation errors across many time steps [13, 14]. The challenges with RNNs in terms of vanishing and exploding gradients are well documented, and a number of approaches to address these problems have been proposed. Long Short Term Memory (LSTM) is an RNN that avoids the effects of vanishing and exploding gradient by controlling how the nodes handle data [15]. LSTM adds controllable gates that instruct a node to allow input, block output, and optionally forget current state, thereby avoiding the effect of vanishing and exploding gradients.

By far the most popular training principle to avoid vanishing and exploding gradients is the maximum likelihood principle. In the RNN literature, this is known as Teacher Forcing [16]. Teacher Forcing refers to using the output of the previous step as input to the current. It has been shown that in practice it can reduce the chance that gradients explode, and even allow training generator models or models that work with unbounded amounts of memory [17]. Other approaches have been suggested and tested as well. Pascanu et al. [18] proposed a gradient norm clipping strategy to deal with exploding gradients and a soft constraint for the vanishing gradients problem. Recently Goyal et al. [19] introduced Professor Forcing, a training model that trains a discriminator to distinguish between sequences generated using Teacher Forcing and scheduled sampling (ordinary updating done to the context nodes). The approach should according to the authors, increase the RNNs ability to model long-term dependencies.

B. Research Questions

In terms of predicting or forecasting what is about to happen some important questions will be addressed:

- How far in advance does the forecasting approach predict the fall?
- Is it possible to quantify how long the intervention period will be based on the current context?
- Do the conditional probabilities of a prediction correspond to the current contexts fall risk?

These questions are addressed by designing and performing an experiment where we use recorded data from an earlier study [20] and extract features from this dataset. The dataset is applied to four different approaches to prediction and forecasting, and evaluate and discuss the findings. Based on the findings the research questions are addressed.

II. METHODS AND MODELS

The approach presented is using data collected during an experiment set up at the UiT nursing school in Narvik, Norway. In [20] Danielsen describes the sensory setup used, the experiment executed and results in terms of recognizing bedside falls. The raw data was captured from the sensors once every second and identified as a frame. Each frame consisted of 4801 distinct values; 4800 thermal readings representing an 80×60 thermal array, and a single distance reading. The distance reading was in the form of centimeters from the ceiling mounted device to the closest reflecting object, while thermal readings were represented as integer numbers. Each integer represented a reading from a single point of the 80×60 thermal sensor with a sensitivity of $0.05 \text{ }^\circ\text{C}$.

28 recordings were made during the experiment by seven participants, three women and four men, all young and healthy. Each recording lasted between 3 and 5 minutes. Time between infrared frames, Δt , was 1 second. The recording resulted in 8032 frames, a total recording time around 2 hours 14 minutes. The bed used during the recordings had full bed linen and some of the participants used it. The participants were instructed to perform a number of activities, but not specifically how to perform the activity. Table 1 gives an overview of the instructions given. The bed position was altered between recordings. More information on the experimental setup, sensors, equipment used, etc. can be found in [20] and in [21].

TABLE I. EXPERIMENT SCENES AND INSTRUCTIONS

Scene 1	Scene 2	Scene 3	Scene 4
Enter room Sit down (in bed)	Enter room Fall down (beside bed)	Enter room Lay down (in bed)	Enter room Lay down (in bed)
Lay down (in bed)	Rise up (from floor) but fail	Change poses (in bed)	Change poses (in bed)
Change poses (in bed) randomly	Sit/lay down (on floor) Change poses	Slip down (on floor) Change poses	Sit up (on bed/bedrail) Change poses
Sit up (on bedrail) Change poses			Fall down when trying to stand up
Stand up (on floor) walk around			Sit/lay down (on floor)

Fig. 1 shows the bed in respect to the sensors during the experiment. The bed used was an ordinary adjustable hospital bed with rails. The bed was altered into three positions during the experiment. The positions are shown in Fig. 1 along with

the bed centering coordinate system orientation used by this paper.

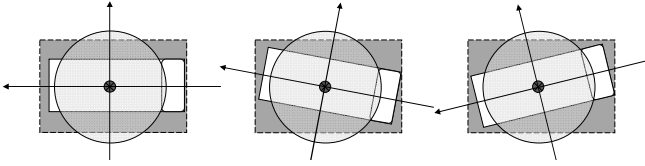


Fig. 1. Sensor registration area and bed center coordination orientation.

The dark point over the bed marks the location of the ceiling-mounted sensors. The sensor registers thermal readings within the square area outlined with a dashed line, while the ultrasonic sensor registers distance readings within the semi-transparent circular area.

A. State Transitions

The approach in [20] was based on a scenario in which the individual being monitored was alone within the observable area. Fig. 2 shows the state-transition diagram for a single individual in a bedroom within the observable area. In the figure, the location is separated by dashed lines. The postures an individual may be in are defined by the oval symbols. The solid arrows between postures illustrate how states involving location and posture change. The ‘‘Sitting’’ and ‘‘Lying’’ posture on the ‘‘Floor’’ is considered the final location and posture after a fall. These states may be entered from any other states.

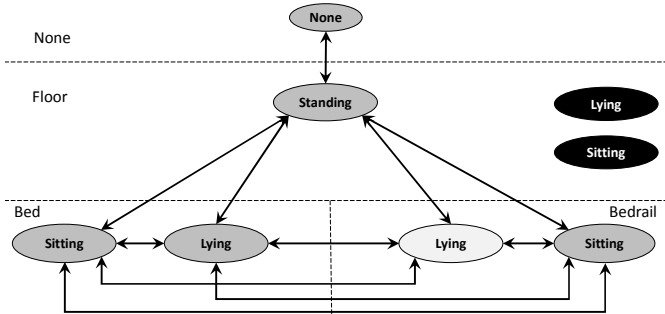


Fig. 2. State transition between location and postures.

B. Feature Extraction

In [21] Danielsen et al presented how background heat and residual heat can be removed from a thermal image. The term ‘‘heat imprint’’ is used to label a frame that has been subjected to both the background subtraction algorithm and the residual heat removal algorithm presented in [21]. In [20] Danielsen used a limited number of features to determine location and posture using the raw data collected during the experiment. In this paper, we have expanded the number of features into two distinct sets of features; static and dynamic frame features.

The static features, SF_f , extracted from a single frame f are presented in Table 2, while the dynamic features, DF_f , calculated by comparing the previous frame with the current are presented in Table 3.

TABLE II. STATIC FRAME FEATURES

Name	Feature within a single frame f
$HCIB_f$	Number of heat imprint pixels left by individual in bed
$HCOB_f$	Number of heat imprint pixels left by individual out of bed
$MTIB_f$	Maximum temperature in bed
$MTOB_f$	Maximum temperature outside bed
D_f	Distance from sensor mounting point to closest reflecting object
DV_f	Boolean indicator: Is distance reading valid due to heat imprint being present within sensor registration area, see Fig. 1.
$CGHIB_f$	Is Center of Gravity (COG) in terms of the heat imprint pixels within the boundaries of the bed?
$CGHOF_f$	Is COG in terms of the heat imprint pixels on the floor?
$CGHDB_f$	Distance from COG of heat imprint to center of bed
$CGHDBR_f$	Distance from COG of heat imprint to the closest bedrail
$CGBIB_f$	Is COG in terms of the bounding box (BB) of the heat imprint pixels within the boundaries of the bed?
$CGBOF_f$	Is COG in terms of the BB heat imprint pixels on the floor?
$CGBDB_f$	Distance from COG of BB heat imprint to center of bed
$CGBDBR_f$	Distance from COG of BB heat imprint to the closest bedrail

TABLE III. DYNAMIC FRAME FEATURES

Name	Feature extracted between frame f and $f-1$
$CGHV_f$	Velocity of COG of heat imprint
$CGHD_f$	Direction in which COG of heat imprint have moved
$CGBV_f$	Velocity of the BBs COG of heat imprint
$CGBD_f$	Direction of the BBs COG of heat imprint

The features presented in Table 2 and 3 were classified using the Random Forest implementation by Breiman [22]. The Random Forest classification approach was chosen specifically for this purpose to reduce possible bias. The classification process results in a probability matrix, PM_f , for each frame f in the dataset, representing the probability of the current frame to be classified as combination of location/posture as defined in Fig. 2, expressed in (1).

$$PM_f = \begin{cases} P_{None}(f) \\ P_{FloorStanding}(f) \\ P_{FloorSitting}(f) \\ P_{FloorLaying}(f) \\ P_{BedrailSitting}(f) \\ P_{BedrailLaying}(f) \\ P_{BedSitting}(f) \\ P_{BedLaying}(f) \end{cases} \quad (1)$$

C. Evaluation Metrics

The ultimate goal is to be able to predict an imminent fall using the current context and the history leading up to the current context as input.

The probability matrix, PM_f , of a single frame f represent how probable the frame is to be classified as a single combination of a location and posture. The probabilities are floating point numbers between 0 and 1, and the sum of all estimated probabilities is 1 within each matrix.

The following metrics have been applied evaluating fall prediction:

- Fall recognition. Is the fall recognized after it has occurred?
- Timeframe window. How far in advance does the approach predict the fall? E.g., how big/small is the window of opportunity to alter the fall risk? Is it possible to give an estimate using the current data?
- False positives – falls that are predicted but do not occur. How do the false positives of the approach reflect an increased fall risk? E.g., do the conditional probabilities calculated by the approach correspond to fall risk?
- True positives – falls that are predicted and do occur. How good is the recognition rate, and what is the timeframe for intervention? Is it possible to determine the timeframe in advance?
- False negatives – falls not recognized until after the fall has occurred. What categorizes the falls that are not recognized?

Even though fall prediction is targeted at avoiding falls from happening, it is important that all falls, independent of when they happen, be detected. Secondly, the timeframe window available for intervention should be as large as possible. Finally, it is not the actual number of false or true positives and false negatives that determine whether an approach is tuned for the task. It is rather the point of whether false positives actually indicate an increased risk of fall or not, and if the false negatives (i.e. falls that happened but where not forecasted) actually were possible to detect before they actually happened.

D. A Sliding Window and its Data

Prediction based on time series imply that the number of previous frames to be included in the time series need to be addressed. By analyzing the sequences of probability matrices generated by the Random Forest classification process during falls, it could be observed that a fall is a process in which the probability of an individual being localized on the floor in a sitting or lying posture increases while the probability of classification of the previous location and posture declines. This process may take some time in the sense that a fall is not necessarily an event that occurs in between frames/seconds. Further it could be observed that the probability of being localized on the floor in a lying or sitting posture was relatively

low most of the time, but raised quickly within three frames if a fall occurred. Based on these observations we defined the frame sequence to be used to consist of three frames to make forecasting probable for the approaches that lack the notion of memory as found in RNNs. To avoid or reduce the problem of exploding and vanishing gradients in RNNs, the classified location and posture of the current frame was added. In essence, a variant of Teacher Forcing [16]. Fig. 3 presents the prediction setup.

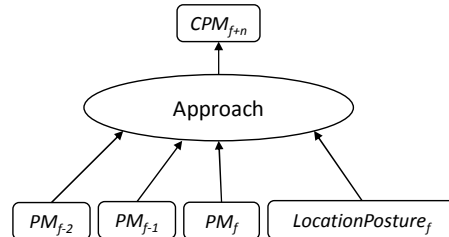


Fig. 3. Predicting the future combination, $f+n$, of location and posture.

The output from the different approaches is a probability matrix that gives the conditional probability for future combination of location and posture. We call this the Conditional Probability Matrix, CPM_{f+n} , where n addresses a future frame relative to f . We define the conditional probability function FA so that for any forecasting approach the conditional probability is determined by three consecutive probability matrices and the current location/posture classification of the current frame f . The FA -function is different for every approach.

$$CPM_{f+n} = FA(PM_{f-2}, PM_{f-1}, PM_f, LocationPosture_f) \quad (2)$$

E. Selected Approaches

When analyzing the data from the Random Forest classifications we observed that a fall happened within a relatively few seconds. Secondly, the falls were obviously dependent on context in respect to where and what was happening and the need of long-term memory in respect to the experiment was not found to be significant. As a result, Elman-RNN was selected because it would demonstrate the concept without the complexity of the LSTM. In addition, neither Markov model or Kalman filter were targeted for benchmarking.

The testing of the approaches was executed using four different artificial intelligence prediction approaches; linear ridge regression, logistic ridge regression, Bayesian Net, and Elman RNN. The selected approaches had the following characteristics:

- Linear regression using a ridge estimator. Attribute selection done by stepping through the attributes removing the ones with the smallest standardized coefficient until no improvement was observable in error estimates.
- Logistic regression using a ridge estimator.
- Bayesian network using the K2 learning algorithm [23] and direct estimates of conditional probabilities

- Elman RNN based on a Multilayer Perceptron (MLP) with 60 nodes in hidden and context layers. The MLP was trained with a learning rate of 0.2 and momentum of 0.1.

F. Defining Fall Risk

A fall state occurs when location/posture is recognized to be Floor/Sitting or Floor/Lying. Any other combination of location/posture is a non-fall state. We use the Fall State Probability, FSP , as a function on any probability matrix M to return the probability in the matrix M of a fall state. The fall state, FS_f , in frame f is defined to be the fall state probability, FSP , of the probability matrix, PM_f , of frame f .

$$FSP(M) = P_{FloorSitting}(M) + P_{FloorLying}(M) \quad (3)$$

$$FS_f = FSP(PM_f)$$

The conditional probability of the future fall state, CFS_{f+n} , is the sum of the conditional probability of Floor/Sitting and Floor/Lying. For approaches that do not maintain a notion of memory, it is defined as:

$$CFS_{f+n} = FSP(CPM_{f+n}) \quad (4)$$

If the forecasting/prediction approach maintains the notion of memory, the CFS_{f+n} is a function of all past states.

$$CFS_{f+n} = P(FSP(CPM_1), FSP(CPM_2), \dots, FSP(CPM_{f+n-1}), FSP(CPM_{f+n})) \quad (5)$$

$$= \prod_{t=1}^{f+n} P(FSP(CPM_t) | FSP(CPM_{t-1}), FSP(CPM_{t-2}), \dots, FSP(CPM_1))$$

The CFS_{f+n} express the conditional probability of a fall state that is about to occur or has occurred. This does not, however, express the future fall risk. The fall risk between the current and the predicted is an expression that quantifies the difference between the fall state probabilities of the current frame, FS_f , and the predicted fall state probability, CFS_{f+n} . We call this property the future fall risk, FR_{f+n} . If a fall has occurred and the FS_f indicate a fall state, the FR_{f+n} is 0. We address this by using a function G returning 0 if the FS_f and the $LocationPosture_f$ both indicate that the individual is located on the floor in a sitting or lying posture, otherwise G returns 1.

$$Val = G(FSP(PM_f), LocationPosture_f) \quad (6)$$

$$FR_{f+n} = Val \times CFS_{f+n}$$

FR_{f+n} express the future fall risk as a number between 0 and 1. This, however, does not imply that the conditional probabilities generated by the different approaches are comparable. To be able to do a comparison in terms of how good the predicted fall risk is, a metric had to be defined deciding when a predicted fall risk should be considered.

G. Preparing and processing the data

Feature extraction, see Table 2 and Table 3, was executed on all 8032 frames. The resulting set was manually labeled with the observed location and posture. 25% of the dataset, 2008 frames, was used for learning purposes to create the Random Forest representation. The Random Forest representation was tested on the rest of the dataset, 6024 records, with a recognition rate of 88.2%.

The Random Forest model was used on the complete dataset to create a probability matrix, PM_f , for each frame in the dataset. The probability matrices were further reorganized according to Fig. 3, and labeled with the observed location and posture of the next record in the dataset. The two first frames in each recording were omitted from the dataset used for prediction. Consequently, the size of the dataset was reduced to 7976 records. 25% of the dataset was used for learning purposes, 1994 records, on each approach.

III. RESULTS

Comparing the four different approaches in terms of how well they perform using the metrics defined, involved defining a threshold value from which triggering of the fall risk, FR_{f+n} , should commence. Based on the results, the threshold value was set to 0.1.

Table 4 show the results from running all four approaches using the applied metrics. The «Time Window» column indicates the time between raised fall risk and the observed fall (true positives). The column gives both the average prediction time for all predicted events, and the minimum and maximum prediction time observed.

A. Recognizing Falls

All four approaches were able to recognize all falls within 1-3 seconds after the fall. This was expected due to the recognition rate of 88.2% using the Random Forest approach toward classifying the location and posture.

TABLE IV. BENCHMARKING RESULTS

Approach	Falls rec.	Time Window	False pos.	True pos.	False neg.	True neg.	Rec. rate
Linear (ridge) Regression	100%	3,8 sec. average 1-22 sec.	17	19	7	7933	95.7%
Logistic (ridge) Regression	100%	1,7 sec. average 1-4 sec.	5	18	8	7945	94.7%
Bayesian Network	100%	1,2 sec.	0	13	13	7950	96.1%
Elman RNN	100%	4,5 sec. average 1-23 sec.	13	22	4	7937	95.8%

B. Performance

The time between detecting the increased fall risk and when the actual fall happened in the approaches, gives some indication of their predictive abilities.

1) Unpredicted Falls (False Negative)

26 falls occur in the dataset. Four falls in the dataset were unpredictable due to lack of information. An additional three falls had similar properties that made classification of location/posture(s) challenging. None of the four approaches predicted any of the four “unpredictable falls”. All approaches did however recognize the “unpredictable falls” immediately after they had occurred.

2) Predictable Falls (True Positive)

Other sequences had very good predictability observable in all approaches. Fig. 4 shows an example on how the future conditional probability fall state, CFS_{f+n} , of the approaches evolve along with the fall state probability, FS_f , of the current frame f , when an individual is rising up from bed and falls. The boxed grey area indicates a fall state. The Elman RNN registered a raised fall risk in frame 172, closely followed by the linear ridge regression approach. The Bayesian network and the logistic ridge regression approaches register a raised fall risk in frame 177 while the fall happened between frame 178 and 179.

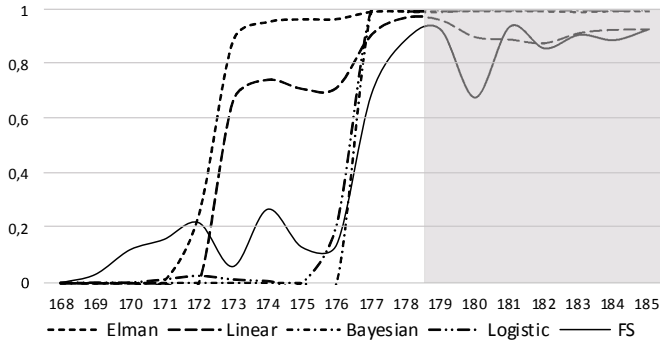


Fig. 4. CFS_{f+n} and FS_f of a fall predicted by all approaches

Fig. 5 shows another fall sequence with different predictability by the approaches. In this sequence, the fall occurs in frame 177. The Elman RNN predicts the fall in frame 174, the logistic ridge regression in frame 175, while the linear ridge regression and Bayesian network fails to predict the fall.

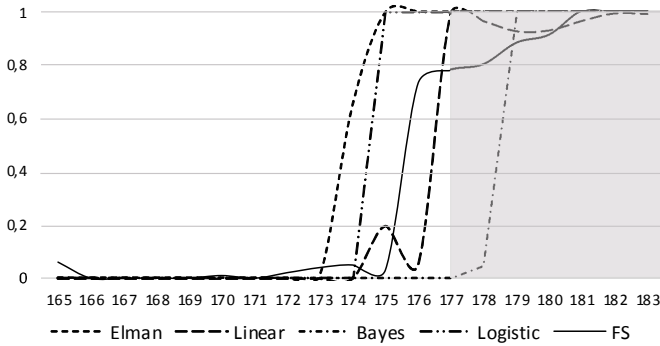


Fig. 5. CFS_{f+n} and FS_f of a fall where some approaches fails to predict fall.

Elman RNN performs better in terms of prediction than the other approaches overall. Analyzing the prediction prior to a fall, it is evident that the Linear regression, Logistic regression, and the Elman RNN approach outperforms the Bayesian network in terms of the number of predictions. Elman RNN and the Linear regression are also on average predicting the fall prior to the Logistic regression by as much as 2-3 sec.

3) Increased Fall Risk, but no Fall (False Positive)

The false positives indicate that the prediction signals an imminent fall, i.e. the future fall risk, FR_{f+n} , has increased sufficiently. However, further processing has revealed that the fall did not occur after all.

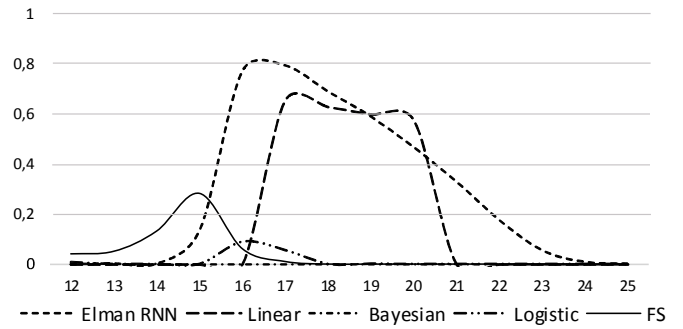


Fig. 6. Fall risk, FR_{f+n} , by approach and the current fall state probability, FS_f .

The question in this context is how good is FR_{f+n} as an indicator of a future fall? Fig. 6 shows the development of the fall risk FR_{f+n} during a sequence of 16 frames. The individual sits down on the bed in frame 14. In frame 15 the individual raised the body up from the bed supporting himself on his arms, and repositioned himself on the bed while slightly turning towards his right side in frame 16. In frame 17 to 25 the individual sits relatively still. The increased FR_{f+n} was registered by the Elman RNN in frame 15 when the individual raised the body up from the bed. At this point, the Elman RNN interpreted the FR_{f+n} to be 0.13, e.g. an increased fall risk. Further movement in frame 16 and 17 amplified the change whereas the FR_{f+n} reduced in the rest of the frames where the individual only sat on the bed without moving much.

Explaining why the linear regression approach behaves as it does is more difficult. The linear regression sets the FR_{f+n} to be 0.66 in frame 17 and keeps the value on approximately this level for four frames while the individual in the frames does not move.

Investigating all false positives in the dataset yields similar results. Explaining why the Elman RNN increase the FR_{f+n} is possible and it is possible to explain why FR_{f+n} is being altered. Similar explanations for the linear regression approach is however more difficult. In addition, the linear regression also experience more false positives.

C. Adding noise to the dataset

The data used in the experiment had very high recognition rates due to the use of Random Forest for classification of location/posture and the conditional probability matrix representing the location/posture in a frame. In [20] Danielsen et al. used an approach that only involved a number of the parameters used here. We recreated the data used in [20] and applied them to the approach presented here. The Random Forest do in this case produce a recognition rate of 86.4% when using 25% of the dataset in teaching. We applied the probability matrices as documented to verify our findings using the Elman RNN approach.

First of all, all falls were recognized. Secondly, the run predicted the same falls (true positives), the run was unable to recognize the “unpredictable falls”, and finally the number of raised alarms in term of increased fall risk was reduced from 13 to 6. The time window was reduced from 4.5 to 2.9. The threshold for triggering a fall risk was set to 0.5. The results are presented in Table 5 and 6 along with the other approaches.

D. Evaluating Results

The evaluated approaches are all able to recognize a fall after a fall has occurred. The Elman RNN approach performed better in terms of how far in advance the fall was predicted. In Table 5 we classify the predictions as short (1-2 seconds), medium (3-5 seconds), and long (> 5 seconds). FR_{f+n} seconds indicate the total number of seconds with a raised fall risk before experiencing a fall.

TABLE V. FORECASTING ON OCCURRED FALLS

	Elman RNN	Linear ridge regression	Logistic ridge regression	Bayes net	Elman RNN ^a
1-2 sec.	8	11	14	11	14
3-5 sec.	8	6	3	2	6
> 5 sec.	6	2	1	0	1
Unpredicted	4	7	8	13	5
FR_{f+n} sec.	98	72	31	17	63

^a. Noise added to dataset by using the data from [21] as described in section C. Adding noise to the dataset

In Table 5, it is evident that the Elman RNN give a medium to long prediction window on 14 of the 22 predictable falls as opposed to linear regression with 8 out of 22 and logistic regression with 4.

Table 6 gives information on situations where a raised fall risk, FR_{f+n} , was registered, but where a fall did not occur. The number of significantly raised fall risks in the dataset is based on human interpretation of all 8032 frames. In addition to the 26 falls, 7 situations were interpreted to have a significantly raised fall risk. The ‘‘Fall risks’’ row shows the number of fall risks recognized by the approaches, while the ‘‘Avg.span’’ shows the average duration of each raised fall risk, while ‘‘Min.’’ and ‘‘Max. span’’ indicates minimum and maximum duration of a raised fall risk in the dataset. ‘‘Recognized’’ shows how many of the 7 raised fall risk situations were recognized by the approach. FR_{f+n} seconds indicate the total number of seconds with a raised fall risk while not experiencing a fall.

TABLE VI. RAISED FALL RISK, BUT NOT FALLING

	Elman RNN	Linear ridge regression	Logistic ridge regression	Bayes net	Elman RNN ^b
Fall risks	13	17	5	0	6
Avg. span	11	6	1	-	1
Min. span	1	1	1	-	1
Max. span	23	20	3	-	1
Recognized	5	4	2	-	4
FR_{f+n} sec.	137	106	10	-	6

^b. Noise added to dataset by using the data from [21] as described in section C. Adding noise to the dataset

The entropy in the dataset used when comparing fall states with non-fall states was 0.9653. Out of the dataset of 7976 records, 3117 are fall state records, while 4859 are non-fall state.

E. Receiver Operating Characteristics

The differences between the observed events of the raw series and the predicted events obtained through the four approaches have been compared to determine the efficacy of the four forecasting approaches in terms of prediction abilities. We have applied receiver operating characteristic (ROC) analysis [24] for this purpose.

Performance in this context is defined as the ability to predict a future fall within a period of 10 seconds. The receiver operating characteristic (ROC) curve presented in Fig. 7 shows performance of each of the four approaches in terms of the probability of predicting an oncoming fall (True Positive Rate) as opposed to the probability of false alarms (False Positive Rate). The dashed diagonal line shows a random guess distribution.

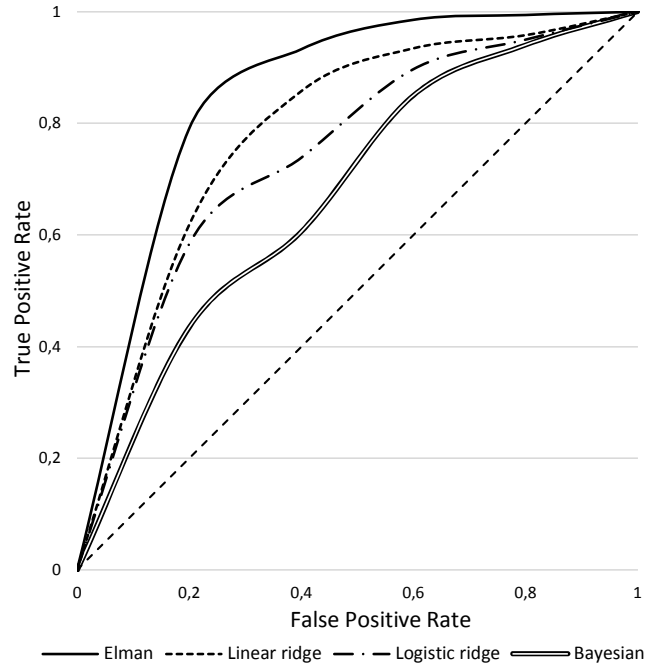


Fig. 7. ROC curve –fall prediction performance within a 10 second window.

When calculating the Area Under Curve (AUC) for the approaches presented in Fig. 7, Elman performs approximately 9% better than linear regression, with a forecasting recognition rate of 85%.

The presented approach towards creating a fall risk indicator using current context has promising properties as well. The ROC curve in Fig. 8 shows the performance of the proposed fall risk indicator, FR_{f+n} , of the four forecasting approaches. The performance in this context is defined as the ability to detect a raised fall risk using current context as opposed to an interpretation of fall risk based on observation. The evaluation may be biased based on the subjective interpretation. The bias is limited since the data used to analyze includes the falls predicted as well.

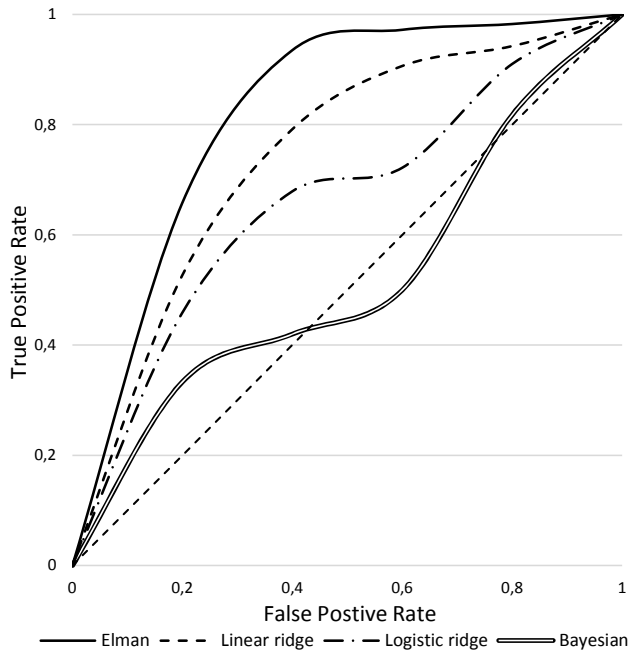


Fig. 8. ROC curve –fall risk performance.

Calculating AUC of Fig. 8 shows that by using the fall risk indicator, FR_{f+n} on each frame f , the Elman RNN performs about 11% better than the Linear ridge regression, 24% better than Logistic regression, and 57% better than the Bayesian approach. The Elman RNN AUC is approximately on 81%.

IV. DISCUSSION

The results from the experiment show that of the selected approaches Elman RNN is best suited for prediction of bedside falls, predicting 22 out of 26 falls using the approach presented, a prediction rate of 84.6%. The Elman RNN offers best prediction rate of both falls and raised fall risks, see Table 4 and 6, and Fig. 7 and 8. In addition, the Elman RNN offers a longer intervention window for altering fall risk and possibly avoiding fall, see Table 5 and 6.

The results in terms of recognizing falls prior to the actual fall are satisfying using the Elman RNN approach. The linear ridge regression performed in most cases nearly as well as the Elman RNN approach. This indicates that some kind of linearity should be observable in the dataset. However, if such linearity was dominant, the Bayesian approach should produce comparable results with linear regression. The Bayesian however performs poorly on all metrics compared to the other approaches.

All falls recognized by the linear ridge regression was recognized by the Elman RNN approach with a better forecasting window, as was the case with the Elman RNN approach in respect to the logistic ridge regression. Linear and logistic ridge regression forecast is based on a single conditional probability matrix, CPM_{f+n} , and the window it represents. They maintain no notion of memory or state in between steps. In Fig. 4 linear ridge regression performs close

to the Elman RNN, but in Fig. 5 the linear ridge regression is unable to predict the fall while the logistic ridge regression performs close to the Elman RNN. The behavior illustrated by Fig. 4 and 5 is observable in all falls, one way or the other. One plausible explanation to this is the notion of memory/state that is preserved by the Elman RNN. It might be able to accumulate the variation in the probability matrices and interpret these, resulting in situations where the Elman RNN triggers a raised fall risk prior to other approaches. This is also what is expressed in equation (4) and (5) in terms of the conditional probability of the future fall state, CFS_{f+n} .

Giving an estimate on how far in advance the fall risk is raised in respect to a potential imminent fall is difficult. This will depend on the individual being monitored, the living environment as well as other parameters.

In 2009 Dykes et al. [25] reported on a larger six-month study showing a positive correlation between the actual number of falls and the awareness of fall risk, both in hospital settings and intervention settings. By raising the awareness of fall risk of the individuals, the number of falls was reduced. Danielsen et al. [26] provides a discussion and design of a fall risk awareness protocol (FRAP) that is suitable in this context. The FRAP combines data from different sensors, both ambient and wearables, and feeds the data into a fall risk probability engine (FRPE). The FRPE combines the actual readings with historical data and current health information to create a fall risk probability score. The fall risk presented in this paper is as such a contribution to this concept. An example of an approach that might be used to increase fall risk awareness is by implementing an IoT RGB bulb like LIFX Color 1000 [27] into the bedroom environment and alter coloring and illumination based on fall risk.

The experiment itself has been executed using young and healthy students, not elderly (age 65+). Consequently, the analyzed data may not represent the behavioral cues of the elderly. This is the major challenge that would need to be resolved in order conclude that the approach presented here will work when monitoring elderly.

V. CONCLUSIONS AND FUTURE WORK

We have presented a novel approach for predicting bedside falls using current context along with the metrics to do an evaluation. The approach has been evaluated based on experimental data and by applying four established methodologies on prediction and forecasting. The results conclude that of the selected approaches, the Elman RNN is best suited for prediction and forecasting given the experimental data, both in terms of recognition rate, forecasting window, and recognition of fall risk in general. We suspect that the good results using Elman RNN are related to the notion of memory/state that is preserved by the Elman RNN and make it possible to accumulate the variation in the probability matrices and interpret these, resulting in situations where the Elman RNN triggers a raised fall risk prior to other approaches. Future studies will elaborate this conclusion. LSTM has shown very promising results in similar situations, and we plan to evaluate the Elman RNN approach presented here with an LSTM approach.

The data used in the experiment were collected in a controlled experimental environment, using young and healthy participants. The participants were instructed to perform a number of activities, but not specifically how to perform these activities. Still, age, eagerness, and speed of the participants may have biased the data collected and the size of the floating window used for predictions. In essence, the data collected may not represent behavioral cues of the elderly. To address this we are organizing a larger study, involving several municipalities, and recording data from elderly volunteers in their homes and in intervention and hospital settings. The recordings will be performed in the bedroom as well as in the bathroom using a similar sensor setup as presented here.

The findings in this experimental study are promising, but hold questions that need to be answered and elaborated before we are able to conclude that the presented approach to fall prediction and current context fall risk are applicable to the elderly.

REFERENCES

- [1] A.M. Tromp, S.M.F. Pluijm, J.H. Smit, D.J.H. Deeg, L. Bouter, et al., "Fall-risk screening test: A prospective study on predictors for falls in community-dwelling elderly", *J Clin Epidemiol*, vol 54, pp. 837–844, 2001.
- [2] B.H. Alexander, F.R. Rivara, and M.E. Wolf, "The cost and frequency of hospitalization for fall-related injuries in older adults", *Am J Public Health*, vol 82, pp. 1020–1023, 1992.
- [3] D.A. Sterling, J.A. O'Connor, and J.A. Bonadies, "Geriatric falls: injury severity is high and disproportionate to mechanism", *J Trauma*, vol 50, pp. 116–119, 2001.
- [4] "Senior's fall in Canada – Second Report". Public Health Agency of Canada. 2014. [Online]. Available: http://www.phac-aspc.gc.ca/seniors-aines/publications/public/injury-blessure/seniors_falls-chutes_aines/assets/pdf/seniors_falls-chutes_aines-eng.pdf
- [5] T.M. Gill, C.S. Williams, M.E. Tinetti, "Environmental Hazards and the Risk of Nonsyncopal Falls in the Homes of Community-Living Older Persons", *Med Care*, vol 38, pp. 1174–1183, 2000.
- [6] S. Asadi, E. Hadavandi, F. Mehmanpazir, and M.M. Nakhostin, "Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction", *Knowl-Based Syst*, vol. 35, pp. 245–258, 2012.
- [7] G. Corani, "Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning", *Ecol Model*, vol 185, pp. 513–529, 2005.
- [8] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *J Basic Eng-T ASME*, vol 82, pp. 35–45, 1960.
- [9] L. Menegaldo, G. Ferreira, M. Santos, and R. Guerato, "Development and navigation of a mobile robot for floating production storage and offloading ship hull inspection", *IEEE T Ind Electron*, vol 56, pp. 3717–3722, September 2009.
- [10] M. Grewal and A. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present" *IEEE Contr Syst Mag*, vol 30, pp. 69–78, June 2010.
- [11] M.I. Jordan, "Serial order: A parallel distributed processing approach", Technical Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986.
- [12] J.F. Elman, "Finding Structure in Time", *Cognitive Sci*, vol 14, pp. 179–211, 1990.
- [13] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *IEEE T Neural Networ*, vol 5, pp. 157–166, March 1994.
- [14] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies - A field guide to dynamical recurrent neural networks", IEEE Press, 2001.
- [15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Comput*, vol 9, pp. 1735–1780, November 1997.
- [16] R.J. Williams, and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks", *Neural Comput*, vol 1, pp. 270–280, June 1989
- [17] R. Pascanu, and H. Jaeger, "A neurodynamical model for working memory", *Neural Networks*, vol 24, pp. 199–207, March 2011
- [18] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks", in *Proceedings of the 30th International conference on Machine Learning*, vol 28, 2013. pp. III-1310–III-1318.
- [19] A. Goyal, A. Lamb, Y. Zhang, S. Zhang, A. Courville, and Y. Bengio, "Professor Forcing: A New Algorithm for Training Recurrent Networks", In *Proceedings of the 29th Conference on Neural Information Processing Systems*, 2016, pp. 4601–4609.
- [20] A. Danielsen, "Non-intrusive Bedside Event Recognition Using Infrared Array and Ultrasonic Sensor", in *Proceedings of the 10th International Conferences on Ubiquitous Computing and Ambient Intelligence*, 2016, pp. 15–25.
- [21] A. Danielsen, J. Tørresen, "Recognizing Bedside Events using Thermal and Ultrasonic Readings", *Sensors*, vol 17, 1342, 2017.
- [22] L. Breiman, "Random Forest", *Mach Learn*, vol 25, pp. 5–32, 2001.
- [23] G.F. Cooper, and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data", *Mach Learn*, vol 9, pp. 309–347, October 1992
- [24] T. Fawcett, "An introduction to ROC analysis", *Pattern Recogn Lett*, vol 27, pp. 861–874, 2006.
- [25] P.C. Dykes, D.L. Carroll, A. Hurley A., S. Lipsitz S., A. Benoit, et al. "Fall Prevention in Acute Care Hospitals: A Randomized Trial", *JAMA- J Am Med Assoc*, vol 304, pp. 1912–1918, November 2010.
- [26] A. Danielsen, H. Olofsen, B.A. Bremdal, "Increasing fall risk awareness using wearables: A fall risk awareness protocol", *J Biomed Inform*, vo. 63, pp. 184–194, October 2016.
- [27] "LIFX Color 1000 Datasheet". Lifx.com. . 2017. [Online]. Available: <https://www.lifx.com/pages/color-1000-info-sheet>