ShortFuse: Biomedical Time Series Representations in the Presence of Structured Information

Madalina Fiterau[1], Suvrat Bhooshan[1], Jason Fries[1], Charles Bournhonesque[2], Jennifer Hicks[3], Eni Halilaj[3], Christopher Ré[1] and Scott Delp[3]

[1]Computer Science Department, Stanford University
[2]Institute for Computational and Mathematical Engineering, Stanford University
[3]Bioengineering Department, Stanford University

Abstract

In healthcare applications, temporal variables that encode movement, health status, and longitudinal patient evolution are often accompanied by rich structured information such as demographics, diagnostics and medical exam data. However, current methods do not jointly optimize over structured covariates and time series in the feature extraction process. We present ShortFuse, a method that boosts the accuracy of deep learning models for time series by explicitly modeling temporal interactions and dependencies with structured covariates. ShortFuse introduces hybrid convolutional and LSTM cells that incorporate the covariates via weights that are shared across the temporal domain. ShortFuse outperforms competing models by 3% on two biomedical applications – forecasting osteoarthritis-related cartilage degeneration and predicting surgical outcomes for cerebral palsy patients.

1. Introduction

In biomedical applications, time series data frequently co-occur with structured information. These time series vary widely in form and temporal resolution, from high-frequency vital signs to longitudinal health indicators in an electronic medical record to activity monitoring data recorded by accelerometers. Structured covariates, such as patient demographics and measures from clinical examinations, are common and complementary to these time series. While abundant, these data are in many cases challenging to integrate and analyze.

For instance, consider data from patients with cerebral palsy (CP), a condition that affects approximately 3 out of every 1000 children in the U.S. (Bhasin et al., 2006). Cerebral palsy makes walking inefficient and sometimes painful. Musculoskeletal surgeries can improve walking, but outcomes are highly variable. Extensive data is available to aid treatment planning, including gait analysis data that characterizes the motion of each joint (e.g., hip, knee, and ankle) during walking, along with a host of structured data such as strength and flexibility measures and birth history (e.g., number of weeks born premature). At many clinical centers, there are roughly as many structured covariates as time series features, from high resolution gait data to clinical visit records collected over several years. All these interconnected factors are not effectively used to aid treatment planning.

Current methods for analyzing these types of data rely on extensive feature engineering, often modeling the time-series and structured information independently. Standard transformations such as Principal Component Analysis (PCA) can be insufficient for capturing all information in time series, requiring additional feature engineering by domain experts. Traditionally, when methods such as PCA, Multiple Kernel Learning (MKL), Dynamic Time Warping (DTW), neural networks or other transformations are used to extract features from time series, the structured covariates in the datasets have no impact on the learned temporal features. In most biomedical applications, there are interactions and correlations between time series and covariates that we would like to leverage. In the case of cerebral palsy, younger children or those with a more severe neural injury might have different gait features that help predict an appropriate surgical plan.

To address this issue, we introduce ShortFuse, a method that boosts the accuracy of deep learning models for time series by leveraging the structured covariates in the dataset. The key to learning relevant representations is to take into account the specifics of the covariates. For instance, cerebral palsy subjects are seen and treated from toddlerhood to adulthood, and the temporal patterns in the joint motion waveforms will depend on the subject's stage of development – while walking on the toes can be normal in toddlers, it is an abnormality in older children or adults. By definition, the structured information is constant along the temporal domain, so unconstrained parameters on the time axis would translate to an additional intercept term and result in overfitting. Consider that two successive gait cycles for the same subject could result in vastly different representations by modifying the time-varying weights of the covariates, which is why this temporal variation of the weights should be discouraged through parameter sharing. Finally, as numerous covariates are not relevant to the predictive task, there must be some mechanism to discount them.

ShortFuse preserves the sequential structure of time series, explicitly modeling interactions and dependencies with structured covariates, allowing the latter to guide feature learning and improve predictive performance. Our approach introduces specialized neural network structures that we call 'hybrid layers' for fusing structured covariates with time series data. The hybrid layers incorporate structured information as distinct inputs, which are used to parametrize, guide, and enrich the feature representations. The first type of layer uses convolutions parametrized by the covariates, where the weights of the structured covariates are shared across the convolutions. Secondly, we introduce a Long-Term Short-Term Memory (LSTM) hybrid, which shares the covariates and their weights across the cells and uses them in the computation of the input gate, forget gate, state change, and output layer. The LSTM hybrid is thus able, for instance, to adjust the length of the forget window.

We demonstrate the versatility of ShortFuse via two representative applications. Critically, our approach makes no assumptions regarding the structure, dimensionality, or sampling frequency of the time series. We also show that the method is flexible, in that it can be applied to LSTMs or Convolutional Neural Networks (CNNs). For these applications, adding structured covariates boosts the accuracy of a time-series-only deep learning model. In addition, ShortFuse matches or improves on results obtained through feature engineering, achieving state-of-the-art accuracy with no input from domain experts. While the focus here is on biomedical applications, ShortFuse is sufficiently general to extend to non-biomedical domains, for instance financial forecasting and sensor-based classification tasks.

2. Related Work

Several different approaches have been used to featurize time series for integration with structured information. A simple approach is to construct histograms of the time series and operate solely on count data. This is a common approach to extract features about physical activity intensity from accelerometer data, e.g. (Dunlop et al., 2011; Song et al., 2010). Principal Component Analysis (PCA) can summarize signals by extracting the linear combinations that account for most of the variance in the data. Previous investigators have used PCA to extract features from joint motion waveforms measured during walking and running and then appended the principal components to other structured information, e.g. (Astephen et al., 2008; Federolf et al., 2013). Segmentation of periodic signals into intervals is also widely used for the processing of vital signs such as ECG (Keogh et al., 2001, 2004) to extract features such as peak-to-peak variability. Methods that account for time series similarity, such as Dynamic Time Warping (DTW) were previously applied to sensor data from an Inertial Measurement Unit (IMU) for gait recognition (Kale et al., 2003) in combination with age and gender information (Trung et al., 2012) and the study of gait in subjects with Parkinson's disease (Wang et al., 2016b). Multiple Kernel Learning (MKL) (Aiolli and Donini, 2014) has been applied to identify brain regions linked to specific gait patterns. (Zhang et al., 2017). Hand engineered features, such as summary statistics, ranges of values, and spectral data extracted from the signal are also frequently employed (e.g., for accelerometer-based activity data (Lee et al., 2015) and joint motion waveforms (Truong et al., 2011; Fukuchi et al., 2011). All these existing methods for combining covariates with time series are highly specialized for their intended application. Optimizing feature representations over all data, temporal and structured, should improve predictive performance by accounting for the interdependence between temporal and structured data.

Deep learning obviates this need for feature engineering and provides a general method to integrate time series and structured covariates, but approaches of joint optimization over these data are largely unexplored. In the past, CNNs and LSTMs have proven apt at encoding temporal information. RNNs and LSTMs were applied to vital signs (Graves et al., 2013). Deep CNNs have advanced performance on network traffic monitoring (Wang et al., 2016a), financial time series (Borovykh et al., 2017), audio (Zheng et al., 2014) and clinical diagnostics (Razavian and Sontag, 2015). Multiscale or Multiresolution CNNs were recently shown to perform well on time series benchmark tasks (Cui et al., 2016). Encoders were shown to benefit anomaly detection from vehicle sensors (Malhotra et al., 2016).

Given the wide range of available deep learning architectures, one could trivially introduce covariates in the initial layer by replicating them along the temporal dimension, thus obtaining an additional constant sequence for each covariate. This poses multiple problems. LSTM layers learn from variations along the temporal domain, inexistent here. With convolutions, there is no parameter sharing, so the replications are treated as separate inputs. This can easily lead to overfitting due to the introduction of parameters at each time point – not needed as the covariates themselves are constant. Also, there are no shortcut connections that may link covariates to later stages in the network, which restricts the information flow. This misses the opportunity for something akin to skip connections (Sermanet et al., 2013), which can enrich representations by connecting arbitrary levels in the network. ShortFuse overcomes all these issues by jointly learning representations over heterogeneous time series and structured data through the hybrid layers described in detail in the next section.

3. ShortFuse

In the sections below, we discuss the fusion of information from time series and structured data using deep neural networks and introduce the technical contributions of ShortFuse.

3.1 Information Fusion

First, we discuss the cases when fusing time series data and covariates leads to improved predictive performance. We assume that the input data have a set of d structured covariates. S is the design matrix, the structured information in the dataset. X is a fixed-length multivariate time series. Y is the univariate output. For a sample i, we use s_i, x_i and y_i to indicate the covariate vector, time series and label. $x_i \in \mathbb{R}^{n \times t}$, where n is the recorded number of sequences, or time series signals, and t is the number of points in time at which the records were captured. y is an integer representing the class label.

Given that the covariates and the sequences typically record different clinical data, it is expected that a predictive model of Y using both X and S will perform better than using either X or S. If Y is not conditionally independent of S given X, that is $I(Y;S|X) \neq 0$, the covariates contain additional information. . Recent work in nonparametric estimation of conditional mutual information (CMI)(Reddi and Póczos, 2013) makes it possible to perform this test. Similarly, if $I(Y;X|S) \neq 0$, the covariates are also insufficiently informative. Fusion of time series and covariates is recommended when both CMI values are > 0.

The simplest approach to introducing covariates in a deep learning model is to replicate each of them, and append it to the time series. Alternatively, they could be introduced in one of the intermediate layers or only used in the final layer. These choices have considerable impact as the relevant temporal features often depend directly on the structured covariates.

For instance, consider the case of two subjects with osteoarthritis, but different body mass index (BMI) values – one is obese and one is normal weight. The task is to determine whether osteoarthritis will progress in time given the subjects' activity counts tracked by accelerometers and other structured covariates. Given the obese subject's higher level of systemic inflammation, cartilage response to high impact loading will be different. In the subject with normal weight the same types of activity may not contribute to disease progression. Instead, for the healthy subject, the mean or minimum activity intensity is possibly more predictive. Thus, structured information present in the dataset has a direct impact on which features should be learned by the model. In this case, not considering the covariates runs the risk of producing less informative features.

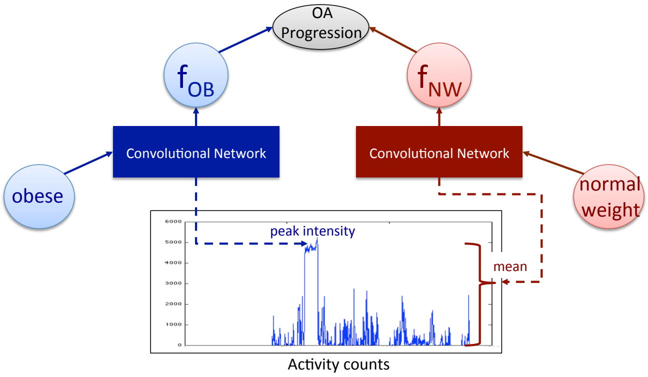


Figure 1: Feature learning mechanism in the presence of covariates. f_{OB} and f_{NW} are internal representations learned by the network. The two parts of the convolutional network learn features relevant for obese subjects and subjects with normal weight, respectively.

Figure 1 illustrates a minimal structure that is capable of leveraging these dependencies to learn and use internal representations as appropriate for the predictive task. Assume there are two binary structured covariates, "obese" and "normal weight", representing subjects' BMI status. If "obese" is active for a subject we use their data to update the first feature f_{OB}. Over time, the feature will become informative in determining whether the subject's osteoarthritis will progress for obese subjects. For instance, it might learn to encode maximum activity intensity. As the covariate "obese" is 0 for subjects with normal weight, feature f_{OB} does not contribute to their predictions. The second feature f_{NW} will be updated in the same way, using data from the subjects with normal BMIs. The internal representations are only influenced by the samples for which they are relevant.

3.2 Fused Architectures

ShortFuse works on the premise that the earlier the covariates are introduced into a given deep network, the more they will be able to direct feature construction. ShortFuse constructs hybrid layers that use the covariates in such a manner that the representational capabilities of LSTMs and CNNs are augmented, meaning that the hypothesis space for the learned features is expanded. They are used predominantly though not exclusively in the initial layers of the network. The key novelty of the hybrid layers is the treatment of structured covariates as global features that are combined with the local temporal patterns encoded by the network. The following section details the hybrid CNNs and LSTMs used to obtain our results, while a complete list, based on commonly used layers, is summarized in Appendix A.

3.2.1 Hybrid Convolutions

The ShortFuse hybrid convolutional layers provide the covariates as parameters to every convolution function along the temporal dimension, together with the time series in that specific window. Figure 2 shows a network with two convolutional hybrid layers.

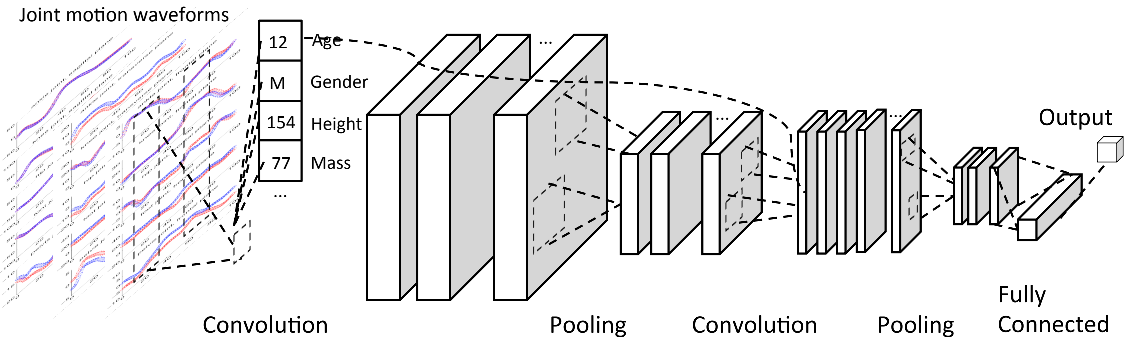


Figure 2: Hybrid convolutional layers using structured covariates. The time series data is shown on the left. The convolutions, with kernels using a subset of the covariates (age, gender, height, and mass), are then applied to the sequences in a time window. There can be several convolutional filters, the outputs of which are pooled, followed by another layer of convolutions which can, in turn, use the covariates. In this example, there is a second pooling layer followed by a fully connected layer and a softmax.

A hybrid convolution maps an input sample x of size n by t to a matrix z of size m by t. Each element $z_{i,j}$ of z is computed from a subset $\bar{x}^{ij}$ of the entries in x.

$$z_{i,j} = \mathbf{1}^T(\bar{x}^{ij} \circ \kappa)\mathbf{1} + \beta \tag{1}$$

where κ is the filter kernel, β is a bias term, $\circ$ denotes the Hadamard product and $\mathbf{1}$ is a vector of ones. Both $\bar{x}^{ij}$ and κ are of size $\bar{n}$ by $\bar{t}$, where $\bar{n} \in [n]$ and $\bar{t} \in [t]$. We use the notation $[u]$ to denote the set $\{1 \ldots u\}$. $\bar{x}^{ij}$ is a submatrix of x determined by selecting $\bar{n}$ signals (rows) from x and the columns of x corresponding to the window of size $\bar{t}$ centered at point j. The m rows of z can be seen as hybrid signals, as each is obtained from a different set of $\bar{n}$ signals in x. For each $i \in [m]$, let V^i be a vector of $\bar{n}$ samples drawn uniformly without replacement from $[n]$. Also, let T^j be the vector of indices of size $\bar{t}$ centered on j.

$$\bar{x}^{ij} = x_{[V^i;T^j]} \tag{2}$$

where the subscripts are the rows and columns from x included in the submatrix.

The convolutions also use the structured covariates and these come into play in the kernel function. Each filter makes use of a randomly selected set of $\bar{d}$ structured covariates, represented by the vector r, the elements of which are drawn uniformly without replacement from $[d]$. We defined κ through parameters w^0 and w as

$$\kappa_{ij} = w^0_{i,j} + \sum_{\ell}^{[\bar{d}]} w_{i,j,\ell} s_{r_\ell} \qquad \forall \quad 0 \leq i \leq [\bar{n}], \quad 0 \leq j \leq [\bar{t}]. \tag{3}$$

We also express the bias as $\beta = \sum_{i=1}^{d} b_i s_i + b^0$, where $b \in \mathbb{R}^d$ and b^0 are parameters.

3.2.2 Hybrid LSTMs

For LSTM-based architectures, the structured covariates are used internally in the computation of the LSTM's nonlinearities, as shown in Figure 3. We introduce weights W_{fs}(forget gate), W_{is} (input gate), W_{Cs}(state change), W_{os} (output gate). The added 's' in the subscript indices of the weights indicate that these weights correspond to the structured covariates s. The terms $W_{\text{fs}} \cdot s$, $W_{\text{is}} \cdot s$, $W_{\text{Cs}} \cdot s$ and $W_{\text{os}} \cdot s$ are added to the arguments of each of the four nonlinearities in the LSTM. The time series values x_{t-1}, x_t and x_{t+1} are provided as input to the cells. The structured covariates s for a given sample are shared across the LSTM cells, together with the covariate weights W_{fs}, W_{is}, W_{Cs} and W_{os}.

3.2.3 LateFuse

A simple alternative to merging time series and covariates data uses a CNN on the structured covariates before the output layer (softmax, binary cross-entropy) of the network. The method is called LateFuse as the covariates are only considered at the end. LateFuse merges the outputs from the network on the time series data and from the covariate CNN.

4. Experimental Design

We developed an evaluation framework to compare ShortFuse to deep learning models that do not use covariates, along with LateFuse and methods that train classifiers on time series representations appended to structured covariates. We selected two representative biomedical applications: identifying candidates for surgical treatment of gait disorders associated with cerebral palsy and predicting cartilage degeneration in patients at risk for osteoarthritis.

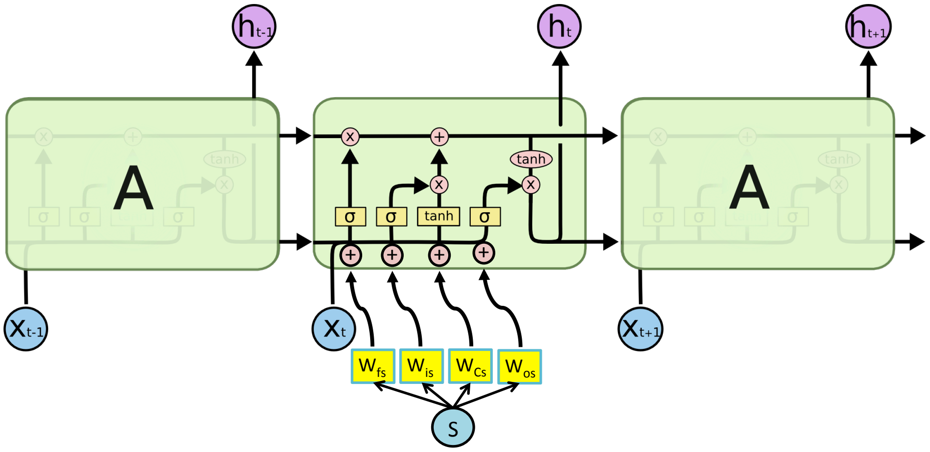


Figure 3: Hybrid LSTM layer. The structured covariate weights are shown in yellow. A dot product between these parameters, shared across cells and the structured covariates is added to the original input to the LSTM nonlinearities. Within a cell, the symbols + and × in the small circles represent binary operations, while tanh in the oval is the activation applied to the output. The functions in the yellow rectangles, σ, σ, tanh and σ represent the nonlinearities of the LSTM for the forget gait, input gate, state change and output gate respectively. The outputs of the LSTM, h_{t-1}, h_t and h_{t+1}, are the learned representations.

4.1 Candidate Models

For each application we tested several deep learning models that have been shown to perform well for time series, as discussed in Section 2. The contenders include an LSTM (Appendix B, Figure 4a) (Graves et al., 2013), a deep CNN (Conneau et al., 2016), a multiresolution CNN (Cui et al., 2016), and a CNN network with an Encoder (Appendix B, Figure 4b) (Malhotra et al., 2016). We compared these CNN and LSTM based models against ShortFuse and LateFuse, using the top-performing deep learning model, hyper-tuned for each application.

We also tested Multiple Kernel Learning (MKL) and Dynamic Time Warping (DTW). MKL is a class of algorithms that uses linear combinations of a few predetermined kernels to predict the output (Aiolli and Donini, 2014). Lambda is a regularization hyperparameter of the MKL algorithm that represents the minimizer of the 2-norm of the vector of distances. DTW is a similarity measure that computes the distance between two time-series. We predict our output using nearest neighbours wherein DTW is used as the distance measure between two samples. We first used these methods with only the time series and then also provided the structured covariates as input by repeating them along the temporal dimension. Another baseline was Random Forests (RF) applied to the top principal components extracted from time series and covariates. Finally, we tested RF on the structured covariates with added features engineered by domain experts from time series data.

4.2 Experimental Protocol

The data were split into training, validation, and test sets to perform hyperparameter tuning and evaluation. We use two-level k-fold cross validation. The outer loop splits the data

90%/10% for training+validation and test, respectively. Model selection is performed within the inner loop, with the 90% set being split once again into $k = 10$ folds, each fold used for validation. We selected the model according to average performance over the validation sets. The hyperparameters with the best validation accuracy are chosen, and the model with these parameters is trained on the 90% training+validation dataset. The range of hyperparameters for each model is in Appendix C, Table 3. The model is evaluated on the 10% test set in the outer loop. We report the average accuracy over the test sets.

5. Discussion of Results

Table 1: Accuracy of the deep learning models on the benchmark datasets. The checkmarks indicate which types of data – covariates or time series – are used by each model. We determined significance by applying the McNemar test, which compares the predictions of each contender against those of ShortFuse.

	Cov.	Time series	**Surgery Outcome Prediction**	**OAI Progression**
Default predictor	-	-	65.25% **	63.37% ****
COV+RF	✓	-	67.42% **	64.57% ****
Engineered Features + RF	✓	✓	78% $^{\Diamond}$	67.10% ****
PCA+COV+RF	✓	✓	72.25% **	67.38% **
MKL	-	✓	63.83% ****	66.46% ****
MKL+COV	✓	✓	76.42% **	68.22% **
DTW	-	✓	71.50% **	71.08% **
DTW+COV	✓	✓	72.33% *	71.54% *
RNN/LSTM	-	✓	67.83% **	69.19% *
Conditional LSTM	✓	✓	74.33% *	72.53% *
Multiresolution CNN	-	✓	74.58% *	71.15% *
Encoder + CNN	-	✓	75.08% *	68.18% **
CNN	-	✓	75.33% *	68.75% **
LateFuse	✓	✓	76.17% **	70.30% **
ShortFuse	✓	✓	**78.92**% [BASE=CNN]	**74.42**% [BASE=LSTM]

$^{\Diamond}$ As obtained by (Schwartz et al., 2013); * $p \in [0.01, 0.05)$ (significant); ** $p \in [0.001, 0.01)$ (very significant); *** $p \in [0.0001, 0.001)$ (extremely significant); **** p<0.0001.

The two biomedical applications, osteoarthritis progression and surgical outcome prediction are described in detail in the next sections. The class imbalance is 63% for osteoarthritis progression and 65% for surgery outcome prediction. The results are summarized in Table 5. For both applications, the RF models trained exclusively on covariates are the worst performers, indicating that time series should be used in the prediction. The results also show that ShortFuse is 3% more accurate than past deep learning models and other methods for automatically learning time series representations. ShortFuse also matches or outperforms models trained by domain experts. We also find that ShortFuse outperforms LateFuse by

2-4% – the structured covariates have a greater impact in the representation learning if they are integrated into the network as opposed to being merged immediately prior to prediction. ShortFuse also outperforms MKL, DTW and PCA+RF by 2-3%, even when these methods use the structured covariates. Unlike in the case of deep learning models, providing covariates to MKL and DTW did not lead to significant increase in accuracy. The hypothesis space expressed by these models is not rich enough to explain the underlying connections between the time series and the structured data.

5.1 Forecasting osteoarthritis progression

Knee osteoarthritis (OA) is a leading cause of disability in older adults (CDC, 2009; Guccione et al., 1994), with 50% of the population at risk of developing symptoms at some point in their life (Murphy et al., 2008). Prevention, which could significantly reduce the burden of this incurable disease, hinges on a deeper understanding of modifiable risk factors, such as physical activity (Dunlop et al., 2014; Lee et al., 2015). Currently, clinicians lack the necessary evidence to make specific activity modification recommendations to patients. Some studies have reported that physical activity is associated with an increased risk of knee OA (Lin et al., 2013; Felson et al., 2013). Others have reported either no association or opposite findings (Racunica et al., 2007; Mansournia et al., 2012). Current suggestions are not fine-tuned to patient demographics, medical histories, and lifestyles. Similar types of activities are expected to have different effects on patients with different joint alignment angles or different levels of systemic inflammation (Griffin and Guilak, 2005). The interaction of these covariates with physical activity is thus important in predicting disease progression. In this example application, our task is to predict the progression of osteoarthritis, in terms of an objective measure of cartilage degeneration called Joint Space Narrowing (JSN).

We use a dataset of 1926 patients collected as part of the Osteoarthritis Initiative (OAI), an ongoing longitudinal observational study on the natural progression of knee OA that monitored patients yearly, collecting medical histories, nutritional information, medication usage, accelerometer-collected physical activity data, and other data from OA-related questionnaires. As part of the study, subjects had radiographs (X-rays) of their knees taken yearly and their activity monitored for one week. Activity time series were provided as activity counts (acceleration time steps per minute). X-ray data had been previously processed to extract the joint space width, or the distance between the thigh and shank bones, which is representative of cartilage thickness. As cartilage degenerates, the joint space becomes narrower. If the decrease in cartilage width is higher than 0.7 mm, the disease is said to have progressed. We used covariates from years 0-4 and physical activity time series from year 4 to predict whether the disease progressed from year 4 to year 6. The structured covariates include 650 clinical features, while the time series represent activity counts obtained over a week of monitoring. The human engineered features are 3-bin histograms, with thresholds established by domain experts to represent light, moderate, and vigorous activity levels.

An RF model that featurizes the activity data using a histogram approach where features are activity totals for 10 bins of activity intensity levels obtains a 67% classification accuracy, which is only slightly above random chance, after accounting for class imbalance. The best base deep learning architecture is LSTM, which we found to perform well for the single-sequence, non-periodic in this application. ShortFuse with a hybrid LSTM obtains an accuracy of 74%, a 7% increase over the histogram and RF approach, a 5% increase over a standard LSTM and a 2% increase over a conditional LSTM.

5.2 Predicting the outcome of surgery in patients with cerebral palsy

Cerebral palsy is a movement disorder caused by damage that occurs to the immature, developing brain, most often before birth. The condition affects 500,000 people in the US (3.3 per 1,000 births), with 8,000 babies and infants diagnosed each year (Bhasin et al., 2006). Automated tools are needed to aid treatment planning and predict surgical outcomes given both the complexity of the disease (patients present with widely varying gait pathologies) and invasive nature of treatments, which include skeletal, muscular, and neural surgeries.

In this application, our task is to predict whether psoas lengthening surgery (a procedure to address a tight or overactive muscle in the pelvic region) will have a positive outcome. As in previous work (Truong et al., 2011), we define a positive outcome as (1) an improvement of more than 5 points in Pelvis and Hip Deviation Index (PHiDI), which is a gait-based measure of dysfunction of motion of the pelvis and hip during walking, or (2) a post-surgical Gait Deviation Index of more than 90, which indicates that the subject's gait pattern is within one standard deviation of a typically-developing child. The time series in the data are joint angles obtained during the subject's gait cycle from motion capture using markers. The computation of the human engineered features requires domain expertise such as knowledge of the stances in the gait cycle (i.e., whether the foot of the limb of interest is in contact with the ground or not). The features are described in Appendix D.

The current state of the art uses an RF model trained on clinical information as well as hand-engineered clinical features, which has an accuracy of 78% (Schwartz et al., 2013). The best performing deep learning architecture is the deep CNN, possibly because the gait time series consists of multiple (15) sequential variables representing joint angles, which all have a shape that does not vary considerably across subjects. ShortFuse improves over the best deep learning model by 3%, matching the performance of the model trained on human engineered features and covariates, thus obviating the need for human designed features.

6. Conclusions

We introduce ShortFuse, a method for incorporating structured covariates into time series deep learning to improve performance over current state-of-the-art models. The key contribution of this work is that the covariates have a direct effect on the representations that are learned, leading to more accurate models. Results indicate that the structured covariates have a greater impact on the representation learning if they are integrated into the network early as opposed to being merged right before the final layer. We have also outperformed other standard baselines, even when the baselines use covariates. Our model outperforms such baselines by 2-3% on two biomedical tasks.

ShortFuse obtains 3% improvement over all other approaches in forecasting osteoarthritis-related cartilage degeneration, 2 years in advance. This is crucial in supporting clinicians in making informed recommendations for patients who present with joint pain. For surgery outcome prediction in cerebral palsy patients, we outperformed or matched the state-of-the-art, at the same time eliminating the need for painstaking feature engineering.

Acknowledgments

The Mobilize Center is a National Institutes of Health Big Data to Knowledge (BD2K) Center of Excellence supported through Grant U54EB020405. Many thanks to Dr. Michael Schwartz at the University of Minnesota and Gillette Children's Specialty Healthcare Hospital for providing the cerebral palsy data. The OAI is a public-private partnership comprised of five contracts (N01-AR-2-2258; N01-AR-2-2259; N01-AR-2-2260; N01-AR-2-2261; N01-AR-2-2262) funded by the National Institutes of Health, a branch of the Department of Health and Human Services, and conducted by the OAI Study Investigators. Private funding partners include Merck Research Laboratories; Novartis Pharmaceuticals Corporation, GlaxoSmithKline; and Pfizer, Inc. Private sector funding for the OAI is managed by the Foundation for the National Institutes of Health. This manuscript was prepared using an OAI public use data set and does not necessarily reflect the opinions or views of the OAI investigators, the NIH, or the private funding partners. We also thank Prof. David Sontag for his helpful suggestions.

References

F. Aiolli and M. Donini. Easy multiple kernel learning. In *22th European Symposium on Artificial Neural Networks (ESANN)*, pages 289–294, 2014.

J. L. Astephen, K. J. Deluzio, G. E. Caldwell, M. J. Dunbar, and C. L. Hubley-Kozey. Gait and neuromuscular pattern changes are associated with differences in knee osteoarthritis severity levels. *Journal of Biomechanics*, 41(4):868–876, 2008.

T. K. Bhasin, S. Brocksen, R. N. Avchen, and K. Van Naarden Braun. *Prevalence of four developmental disabilities among children aged 8 years: Metropolitan Atlanta Developmental Disabilities Surveillance Program, 1996 and 2000*. US Department of Health and Human Services, Centers for Disease Control and Prevention, 2006.

A. Borovykh, S. Bohte, and C. W. Oosterlee. Conditional time series forecasting with convolutional neural networks. *preprint arXiv:1703.04691*, 2017.

CDC. Prevalence and most common causes of disability among adults—United States, 2005. *MMWR: Morbidity and Mortality weekly report*, 58(16):421–426, 2009.

A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun. Very deep convolutional networks for text classification. *preprint arXiv:1606.01781*, 2016.

Z. Cui, W. Chen, and Y. Chen. Multi-scale convolutional neural networks for time series classification. *preprint arXiv:1603.06995*, 2016.

D. D. Dunlop, J. Song, P. A. Semanik, R. W. Chang, L. Sharma, J. M. Bathon, C. B. Eaton, M. C. Hochberg, R. D. Jackson, C. K. Kwoh, et al. Objective physical activity measurement in the osteoarthritis initiative: are guidelines being met? *Arthritis & Rheumatology*, 63(11):3372–3382, 2011.

D. D. Dunlop, J. Song, P. A. Semanik, L. Sharma, J. M. Bathon, C. B. Eaton, M. C. Hochberg, R. D. Jackson, C. K. Kwoh, W. J. Mysiw, et al. Relation of physical activity time to incident disability in community dwelling adults with or at risk of knee arthritis: prospective cohort study. *BMJ*, 348:g2472, 2014.

P. Federolf, K. Boyer, and T. Andriacchi. Application of principal component analysis in clinical gait research: identification of systematic differences between healthy and medial knee-osteoarthritic gait. *Journal of Biomechanics*, 46(13):2173–2178, 2013.

D. T. Felson, J. Niu, T. Yang, J. Torner, C. E. Lewis, P. Aliabadi, B. Sack, L. Sharma, A. Guermazi, J. Goggins, et al. Physical activity, alignment and knee osteoarthritis: data from MOST and the OAI. *Osteoarthritis and cartilage*, 21(6):789–795, 2013.

R. K. Fukuchi, B. M. Eskofier, M. Duarte, and R. Ferber. Support vector machines for detecting age-related changes in running kinematics. *Journal of Biomechanics*, 44(3): 540–542, 2011.

A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649. IEEE, 2013.

T. M. Griffin and F. Guilak. The role of mechanical loading in the onset and progression of osteoarthritis. *Exercise and sport sciences reviews*, 33(4):195–200, 2005.

A. A. Guccione, D. T. Felson, J. J. Anderson, J. M. Anthony, Y. Zhang, P. Wilson, M. Kelly-Hayes, P. A. Wolf, B. E. Kreger, and W. B. Kannel. The effects of specific medical conditions on the functional limitations of elders in the Framingham study. *American Journal of Public Health*, 84(3):351–358, 1994.

A. Kale, N. Cuntoor, B. Yegnanarayana, A. Rajagopalan, and R. Chellappa. Gait analysis for human identification. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 706–714. Springer, 2003.

E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Proceedings, IEEE International Conference on Data Mining (ICDM)*, pages 289–296. IEEE, 2001.

E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 57:1–22, 2004.

J. Lee, R. W. Chang, L. Ehrlich-Jones, C. K. Kwoh, M. Nevitt, P. A. Semanik, L. Sharma, M.-W. Sohn, J. Song, and D. D. Dunlop. Sedentary behavior and physical function: objective evidence from the osteoarthritis initiative. *Arthritis Care & Research*, 67(3): 366–373, 2015.

W. Lin, H. Alizai, G. Joseph, W. Srikhum, M. Nevitt, J. Lynch, C. McCulloch, and T. Link. Physical activity in relation to knee cartilage T2 progression measured with 3T MRI over a period of 4 years: data from the osteoarthritis initiative. *Osteoarthritis and Cartilage*, 21(10):1558–1566, 2013.

P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff. LSTM-based encoder-decoder for multi-sensor anomaly detection. *preprint arXiv:1607.00148*, 2016.

M. A. Mansournia, G. Danaei, M. H. Forouzanfar, M. Mahmoodi, M. Jamali, N. Mansournia, and K. Mohammad. Effect of physical activity on functional performance and knee pain in patients with osteoarthritis: analysis with marginal structural models. *Epidemiology*, 23(4):631–640, 2012.

L. Murphy, T. A. Schwartz, C. G. Helmick, J. B. Renner, G. Tudor, G. Koch, A. Dragomir, W. D. Kalsbeek, G. Luta, and J. M. Jordan. Lifetime risk of symptomatic knee osteoarthritis. *Arthritis Care & Research*, 59(9):1207–1213, 2008.

OAI. Osteoarthritis initiative, a knee health study, 2017. URL `https://oai.epi-ucsf.org/datarelease/`.

T. L. Racunica, A. J. Teichtahl, Y. Wang, A. E. Wluka, D. R. English, G. G. Giles, R. O'Sullivan, and F. M. Cicuttini. Effect of physical activity on articular knee joint structures in community-based adults. *Arthritis Care & Research*, 57(7):1261–1268, 2007.

N. Razavian and D. Sontag. Temporal convolutional neural networks for diagnosis from lab tests. *preprint arXiv:1511.07938*, 2015. URL `http://arxiv.org/abs/1511.07938`.

S. J. Reddi and B. Póczos. Scale invariant conditional dependence measures. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1355–1363, 2013.

M. H. Schwartz, A. Rozumalski, W. Truong, and T. F. Novacheck. Predicting the outcome of intramuscular psoas lengthening in children with cerebral palsy using preoperative gait data and the random forest algorithm. *Gait & Posture*, 37(4):473–479, 2013.

P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3626–3633, 2013.

J. Song, P. Semanik, L. Sharma, R. W. Chang, M. C. Hochberg, W. J. Mysiw, J. M. Bathon, C. B. Eaton, R. Jackson, C. K. Kwoh, et al. Assessing physical activity in persons with knee osteoarthritis using accelerometers: data from the osteoarthritis initiative. *Arthritis Care & Research*, 62(12):1724–1732, 2010.

N. T. Trung, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi. Performance evaluation of gait recognition using the largest inertial sensor-based gait database. In *5th IAPR International Conference on Biometrics (ICB)*, pages 360–366. IEEE, 2012.

W. H. Truong, A. Rozumalski, T. F. Novacheck, C. Beattie, and M. H. Schwartz. Evaluation of conventional selection criteria for psoas lengthening for individuals with cerebral palsy: a retrospective, case-controlled study. *Journal of Pediatric Orthopaedics*, 31(5):534–540, 2011.

W. Wang, C. Chen, W. Wang, P. Rai, and L. Carin. Earliness-aware deep convolutional networks for early time series classification. *preprint arXiv:1611.04578*, 2016a.

X. Wang, M. Kyrarini, D. Ristić-Durrant, M. Spranger, and A. Gräser. Monitoring of gait performance using dynamic time warping on IMU-sensor data. In *IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6. IEEE, 2016b.

Y. Zhang, S. Prasad, A. Kilicarslan, and J. L. Contreras-Vidal. Multiple kernel based region importance learning for neural classification of gait states from EEG signals. *Frontiers in Neuroscience*, 11, 2017.

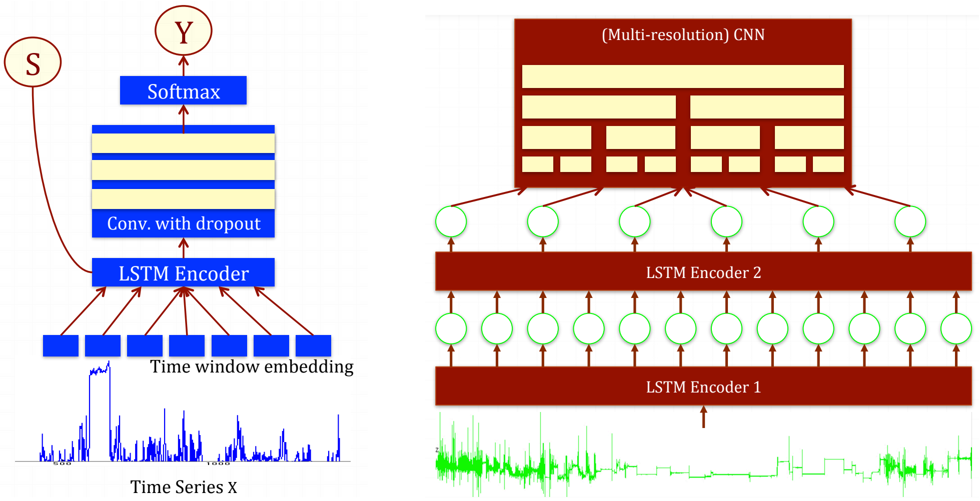
Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao. Time series classification using multi-channels deep convolutional neural networks. In *International Conference on Web-Age Information Management*, pages 298–310. Springer, 2014.

Appendix A. List of hybrid layers

Table 2: List of ShortFuse hybrid layers and connections with standard layers.

Standard Layer	Hybrid Layer
Convolution 1D	Covariates provided to convolutions along temporal dimension.
Concolution 2D	Interleave covariates to obtain a sequence of the same periodicity and size as the time series data.
Fully Connected	Covariates inputted to each one of the fully connected cells.
RNN/LSTM	Use the structured covariates as part of additive terms in the computation of the LSTM parameters.

Appendix B. Figures of deep learning models



(a) LSTM with embeddings.

(b) Encoder + CNN.

Figure 4: Deep learning candidate models.

Appendix C. Model parameters

Table 3: Hyperparameters used in the model training and the models they apply to.

Hyperparameter	Model to which it applies	Parameter range for search
Learning rate	RNN / LSTM / all CNN models	0.001 - 0.003
Dropout	RNN / LSTM / all CNN models	0.0 - 0.5
Embedding size	LSTM	16 - 64
Number of filters	all CNN models	3-13
Number of layers	all CNN models	1-10
Resolutions	Multiresolution CNN	256 - 128 - 64 - 32 -16
Kernel	Multiple Kernel Learning	RBF
Number of trees	Random Forests	10 - 1000

Appendix D: Features of the cerebral palsy dataset

Table 4: Structured covariates used for psoas lengthening outcome prediction.

Feature Name	**Range of values**
Side	Left/Right
Functional Assessment Questionnaire	$[0, 10]$
Gross Motor Function Classification System	$[0, 4]$
Age	$[3.67, 32.91]$
Height	$[93.98, 184.90]$
Mass (kg)	$[12, 74.60]$
BMI	$[11.44, 43.55]$
Leg length	$[43, 101]$
Cadence	$[0.55, 3.64]$
Speed	$[0.0677, 1.7133]$
Steplen	$[0.0702, 0.8761]$
Triplegic	$\{0, 1\}$
Quadriplegic	$\{0, 1\}$
Previous PHiDI	$[68.67, 122.71]$

Appendix E: Benchmark results with p-values

Table 5: Accuracy of the deep learning models on the benchmark datasets. The p-values are obtained using McNemar's test to compare each contender against ShortFuse.

	CP Psoas Prediction	**OAI Progression**
Default predictor	65.25% (p=0.0010)	63.37% (p=2.24e-07)
COV+RF	67.42% (p=0.0066)	64.57% (p=4.18e-06)
Engineered Features + RF	78% *	67.10% (p=5.67e-04)
PCA+COV+RF	72.25% (p=0.0078)	67.38% (p=0.0011)
MKL	63.83% (p=2.22e-04)	66.46% (p=2.80e-04)
MKL+COV	76.42% (p=0.0488)	68.22% (p=0.0031)
DTW	71.50% (p=0.0086)	71.08% (p=0.0098)
DTW+COV	72.33% (p=0.0198)	71.54% (p=0.0165)
RNN/LSTM	67.83% (p=0.0093)	69.19% (p=0.0110)
Conditional LSTM	74.33% (p=0.0235)	72.53% (p=0.0344)
Multiresolution CNN	74.58% (p=0.0272)	71.15% (p=0.0112)
Encoder + CNN	75.08% (p=0.0320)	68.18% (p=0.0031)
CNN	75.33% (p=0.0450)	68.75% (p=0.0061)
LateFuse	76.17% (p=0.0476)	70.30% (p=0.0425)
ShortFuse	**78.92**% [BASE=CNN]	**74.42**% [BASE=LSTM]

* As obtained by (Schwartz et al., 2013).