Received: 2 July 2020 | Accepted: 30 July 2020
DOI: 10.1002/sta4.305

WILEY

SPECIAL ISSUE PAPER

Semi-supervised joint learning for longitudinal clinical events classification using neural network models

Weijing Tang[1] | Jiaqi Ma[2] | Akbar K. Waljee[3] | Ji Zhu[1]

[1]Department of Statistics, University of Michigan, Ann Arbor, Michigan, 48109, USA

[2]School of Information, University of Michigan, Ann Arbor, Michigan, 48109, USA

[3]Michigan Medicine, Department of Internal Medicine, Division of Gastroenterology and Hepatology, University of Michigan, Ann Arbor, Michigan, 48109, USA

Correspondence
Ji Zhu, Department of Statistics, University of Michigan, Ann Arbor, MI 48109, USA.
Email: jizhu@umich.edu

Present Address
Ji Zhu, 323 West Hall, Ann Arbor, MI 48109, USA.

Funding information
National Science Foundation, Grant/Award Number: DMS-1821243

The success of deep learning neural network models often relies on the accessibility of a large number of labelled training data. In many health care settings, however, only a small number of accurately labelled data are available while unlabelled data are abundant. Further, input variables such as clinical events in the medical settings are usually of longitudinal nature, which poses additional challenges. In this paper, we propose a semi-supervised joint learning method for classifying longitudinal clinical events. Specifically, our model consists of a sequence generative model and a label prediction model, and the two parts are learned end to end using both labelled and unlabelled data in a joint manner to obtain better prediction performance. Using five mortality-related classification tasks on the Medical Information Mart for Intensive Care (MIMIC) III database, we demonstrate that the proposed method outperforms the purely supervised method that uses labelled data only and existing two-step semi-supervised methods.

KEYWORDS
joint learning, longitudinal features, recurrent neural networks, semi-supervised learning

1 | INTRODUCTION

Deep neural network models have been increasingly used to analyse large-scale electronic health records (EHRs) and have shown superior prediction performances in several medical tasks including automatic detection of diabetic retinopathy using medical images (Gulshan et al., 2016) and clinical text classification (Yao, Mao, & Luo, 2019). As opposed to medical images and clinicians' text notes, input features such as clinical events are usually of longitudinal nature. Specifically, sensor recordings, laboratory test results, medications, and new diagnosis codes are recorded on each clinical visit and may change over time. Such longitudinal nature is often accompanied by additional modelling challenges such as irregular time gaps between visits, varying lengths of follow-ups, and complex missing patterns. Recurrent neural networks (RNNs), given their clear advantages in taking sequential inputs and successes in natural language processing (NLP) (Wu et al., 2016), are a natural choice for handling longitudinal inputs, and in recent years, they have been successfully used to analyse clinical events data in different applications such as early detection of heart failure (Choi, Schuetz, Stewart, & Sun, 2016), kidney failure after transplantation (Esteban, Staeck, Baier, Yang, & Tresp, 2016), and daily sepsis and myocardial infarction (Kaji et al., 2019).

Despite many existing successful applications of RNNs on the classification of clinical events data, most of them rely on the accessibility of a large number of accurately labelled training data. However, in many health care settings, qualified graders and disease/domain experts are required to make an accurate diagnosis. Moreover, invasive measurements may result in additional risk to patients, and non-invasive measurement may not be ubiquitous and may result in substantial cost. Therefore, it is often difficult to collect a large number of accurate labels, which limits further applications of deep learning models on clinical events data when labels are scarce. On the other hand, with the availability of routinely collected EHR, there usually exists abundant and easy-to-collect unlabelled data. Therefore, our goal is to develop semi-supervised learning methods for longitudinal clinical events, which can incorporate unlabelled data to help improve classification performance. Successful implementation of such methodology will help reduce costs of collecting clinical labels when building prediction models.

Although there have been many works on semi-supervised learning in the field of deep learning (Kingma, Mohamed, Rezende, & Welling, 2014; Narayanaswamy et al., 2017; Odena, 2016; Socher et al., 2013), there are few works that take longitudinal input such as laboratory tests and charted events that are commonly seen in EHR. Further, most existing approaches treat feature extraction using unlabelled data and building prediction models using labelled data as two separate steps (Ballinger et al., 2018; Che, Cheng, Zhai, Sun, & Liu, 2017; Dai & Le, 2015). The potential drawback of such a two-step approach is that the learned feature representation in the first step receives no supervised guidance from labelled data and, therefore, may not be specific to the desired task.

To overcome the lack of supervision in the first step, we propose to jointly learn feature representation from both labelled and unlabelled data. Our model consists of two parts: a sequence generative network for modelling longitudinal clinical events and a label prediction network which takes the hidden feature representation of the sequence generative network as inputs. The two parts are learned end to end using both labelled and unlabelled training data in a joint manner, such that the data could be well separated in the shared feature space. We empirically show that the proposed joint learning method significantly outperforms the two-step method when labels are scarce. Furthermore, we consider two different generative models for modelling longitudinal clinical events. In addition to the RNNs that have been used in the aforementioned works, where all recurrent layers are deterministic, we also adopt stochastic RNNs which contain an additional stochastic latent recurrent layer. Based on our numerical experiments, taking stochastic RNNs as the generative model could further improve the prediction performance in most cases.

The rest of this paper is organized as follows. We introduce related work in Section 2 and present the proposed semi-supervised joint learning approach with technical details in Section 3. We demonstrate the effectiveness of the proposed method in Section 4 and conclude the paper with discussions in Section 5.

2 | RELATED WORK

Many semi-supervised learning methods have been proposed for deep learning models (Dai & Le, 2015; Kingma et al., 2014; Narayanaswamy et al., 2017; Odena, 2016; Socher et al., 2013). In particular, deep generative models have made great progress on learning feature representations with little or no supervised information in recent years (Cho et al., 2014; Chung et al., 2015; Goodfellow et al., 2014; Kingma & Welling, 2014) and have shown their advantages on unsupervised and semi-supervised tasks. For instance, Kingma et al. (2014) proposed a two-step semi-supervised learning method by first learning a low-dimensional feature representation from unlabelled images via variational autoencoder (VAE) (Kingma & Welling, 2014) and then learning an image classifier from labelled data. However, of the semi-supervised learning methods, only a few can be applied to accommodate longitudinal clinical events (Ballinger et al., 2018; Che et al., 2017; Dai & Le, 2015). Among them, Dai and Le (2015) proposed to pre-train parameters in an RNN encoder with large amounts of unlabelled data and then learn specific text classification tasks starting with pre-trained initialization. Other unsupervised representation learning algorithms such as word2vec (Mikolov, Chen, Corrado, & Dean, 2013) can also be used in the pre-training step. They showed that such pre-training procedure using unlabelled data could provide stable initialization and could be generalized well in different text classification tasks. Following this approach, DeepHeart (Ballinger et al., 2018) also pre-trained parameters in RNNs on unsupervised and weakly supervised tasks and then built a prediction model for four conditions associated with cardiovascular risks using labelled data. More recently, the ehrGAN (Che et al., 2017) was developed to generate realistic patients' clinical events via unsupervised learning. Based on the implicit belief that the generated samples from ehrGAN with input x are likely to have the same label as x, they were further used to produce pseudo labelled data for supervised learning. However, this assumption may not hold in general since the learning procedure of ehrGAN in the first step does not use any label information.

All of the aforementioned semi-supervised learning methods for classifying longitudinal clinical events separate the learning process into two steps: (1) learn a deep generative model using unlabelled data to either pre-train the parameters or augment data and (2) learn a classifier for a specific classification task using labelled data based on the pre-trained initialization or augmented labelled data obtained in the first step. The key potential limitation of such two-step methods is that there is no or weak supervision from labels in the first step. Although data points may cluster well in the feature space by learning the intrinsic structure from unlabelled data, the clusters do not necessarily correspond to the labels of interest. The joint learning approach in our proposed method would make use of label information to help learn feature representation that can better separate data corresponding to the labels, and therefore, obtain better prediction performance.

3 | SEMI-SUPERVISED JOINT LEARNING WITH LONGITUDINAL FEATURES VIA NEURAL NETWORKS

In this section, we first describe the problem set-up of semi-supervised classification for longitudinal features, using clinical events as a specific example. Then we introduce the neural network models, and we present the joint learning approach.

3.1 | Problem set-up

We focus on two different types of features that are commonly seen in EHR: longitudinal features and time-static features. Longitudinal features may include multiple laboratory measurements, charted observations, and active treatments. These features are recorded every time a patient comes for a clinical visit or new laboratory tests or medications are ordered. We denote longitudinal features by $x = (x_1, \dots, x_\ell)$, where $x_j \in R^{d_1}$ for $j = 1, \dots, \ell$, and ℓ is the length of the sequence and can be different for different individuals. Time-static features may include gender, race, admission type, and age at enrolment, which are constant throughout the entire study. We denote time-static features by $w \in R^{d_2}$. The label $y \in \{1, \dots, K\}$ could be the corresponding class associated with mortality or progression of diseases. In the semi-supervised learning setting, we observe only a small number of labelled data (x^i, w^i, y^i) for $i = 1, \dots, n$, and a large number of unlabelled data (x^i, w^i) for $i = n+1, \dots, n+m$, where m is usually much greater than n. We aim to learn a classifier that maps (x, w) to a class label y and incorporate both unlabelled and labelled data to improve prediction performance.

3.2 | Model structure

We propose two neural network models whose architectures are given in Figure 1. Each model consists of two parts: (1) a sequence probabilistic generative network for longitudinal features, which takes any sequence of longitudinal features $(x_1, x_2, \dots, x_{j-1})$ as inputs and models the distribution of features at the next time step, i.e. $p(x_j | x_1, \dots, x_{j-1})$; (2) a label prediction network which takes the hidden recurrent layer of the sequence generative model and the time-static features as inputs and outputs the probability for each class.

3.2.1 | Sequence generative network

We consider two different generative models to model what comes next in a sequence. The first one is a gated recurrent unit (GRU) neural network. We choose GRU because it can better capture long-term dependency due to the additional gate mechanisms compared with vanilla RNNs (Cho et al., 2014). The second one is a variational RNN (VRNN), which contains an additional stochastic recurrent layer. It has been shown that introducing a latent stochastic recurrent layer can provide significant improvements in natural speech processing (Chung et al., 2015), and we adopt it here to examine its potential advantages in modelling clinical events. We describe the two probabilistic models in detail below.

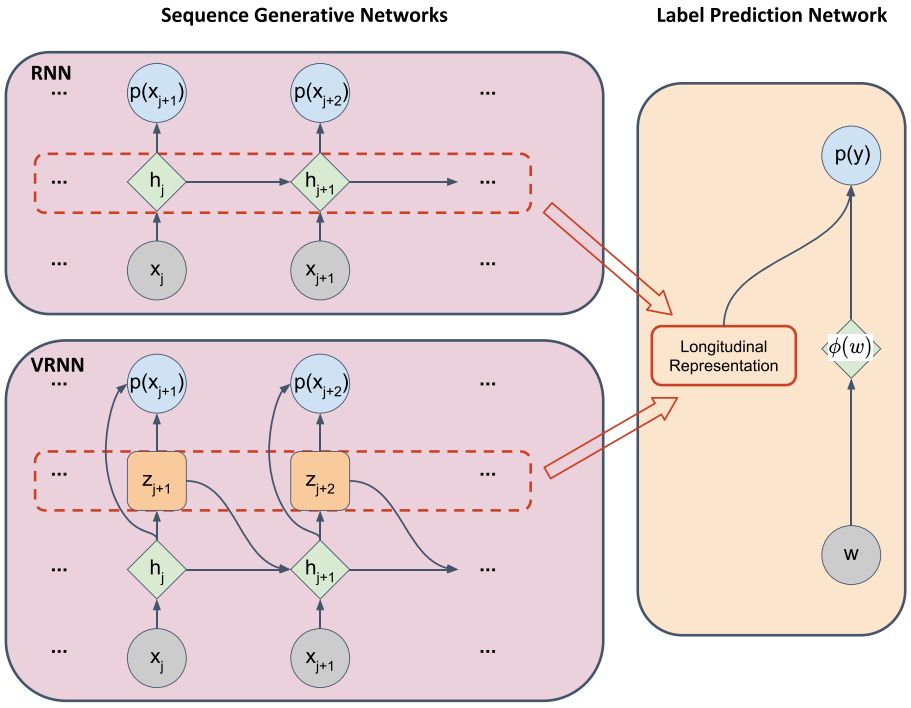

FIGURE 1 Model structure of two sequence generative networks and the label prediction network, where circles represent the inputs and outputs, diamonds represent deterministic hidden layers, and squares represent the stochastic latent recurrent layer in variational recurrent neural network (VRNN).

RNN: As shown in the upper left corner of Figure 1, the hidden units in the recurrent layer $h = (h_1, \dots, h_\ell)$ leverage historical information through the recurrent connection $h_j = f(x_j, h_{j-1})$, where f is a non-linear transformation introduced in GRU. The historical information stored in h_{j-1} determines the distribution of longitudinal features at next time step. Specifically, the conditional density of x_j is given by $p(x_j; h_{j-1}) = \psi(x_j, h_{j-1})$, where ψ is an appropriate density function. For example, if x_j is continuous, we can use a multivariate Gaussian distribution $x_j \sim \mathcal{N}(\mu_{x,j}, \mathrm{diag}(\sigma^2_{x,j}))$, where $[\mu_{x,j}, \sigma_{x,j}] = \xi(h_{j-1})$ and ξ is modelled by a fully connected neural network. Here, we assume different components of x_j are uncorrelated conditional on h_{j-1} in $p(x_j; h_{j-1})$, but they can be correlated in the marginal distribution $p(x_j)$. Since all transformations are deterministic, the joint probability density of longitudinal features is given by

$$p(x) = \prod_{j=1}^{\ell} p(x_j | x_1, \dots, x_{j-1}) = \prod_{j=1}^{\ell} p(x_j; h_{j-1}),$$

where h_0 is usually set as a zero vector in practice.

VRNN: As shown in the bottom left corner of Figure 1, there is an additional stochastic recurrent layer $z = (z_1, \dots, z_\ell)$ compared to the RNN. In particular, layer z is different from the standard hidden layer h since z_j's are random variables while h_j's take deterministic values. The conditional distribution of z_j accesses the historical information through the hidden state h_{j-1}. Specifically, the variable z_j is assumed to follow a multivariate Gaussian distribution with mean $\mu_{z,j}$ and variance $\mathrm{diag}(\sigma^2_{z,j})$, which are determined through a fully connected neural network taking h_{j-1} as inputs. Moreover, the distribution of x_j will be conditioned not only on h_{j-1} but also on the latent z_j, i.e. $p(x_j | z_j; h_{j-1}) = \psi(x_j, \rho(z_j), h_{j-1})$, where ψ is an appropriate density function and ρ is a feature extractor with a two-layer fully connected neural network. Note that, in contrast to the assumption in RNN, now different components of x_j can be correlated conditional on h_{j-1} in $p(x_j; h_{j-1})$, after marginalizing z_j in $p(x_j | z_j; h_{j-1})$. Overall, the joint distribution of longitudinal features x and latent recurrent features z is given by

$$p(x, z) = \prod_{j=1}^{\ell} p(x_j, z_j | x_1, \dots, x_{j-1}, z_1, \dots, z_{j-1}) = \prod_{j=1}^{\ell} p(x_j | z_j; h_{j-1}) p(z_j; h_{j-1}),$$

where, similarly as RNN, h_0 can be set as a zero vector. The hidden units are updated through the recurrence equation $h_j = f(x_j, [z_j, h_{j-1}])$, where f is a GRU module treating the concatenation of z_j and h_j as the hidden state.

3.2.2 | Label prediction network

The label prediction network takes the recurrent layer of the sequence generative network and the time-static features as inputs and returns the probability of belonging to each class. As shown on the right of Figure 1, we first use a feature extractor ϕ for time-static features, where ϕ is a fully connected neural network taking w as inputs. Then we merge the information from both longitudinal features and time-static features by concatenating the hidden feature representation of the sequence generative network and the extracted features $\phi(w)$. Specifically, we utilize the recurrent hidden layer h for RNN and $\tilde{\mu}_z(x) = E_{z \sim q(z|x)} z$, the expectation of the approximate posterior $q(z|x)$ (to be specified later in Section 3.3) of the stochastic recurrent layer z for VRNN. When different individuals have varying lengths of longitudinal features, we can apply a max pooling layer on $h(x)$ or $\tilde{\mu}_z(x)$ over the time steps before we concatenate them with $\phi(w)$. After merging the feature representations, another fully connected neural network along with a Softmax output layer φ is used to output the probability scores, i.e. $p(y|x,w) = \varphi(y; h(x), \phi(w))$ for RNN and $\varphi(y; \tilde{\mu}_z(x), \phi(w))$ for VRNN.

3.3 | Joint learning

The sequence generative network and the label prediction network are learned jointly end to end through shared parameters in the representation of longitudinal features. Specifically, we minimize an objective function that consists of an unsupervised loss and a supervised loss.

The unsupervised loss is constructed by using the negative log-likelihood for longitudinal features x. For RNN, the unsupervised loss is given by

$$\mathcal{L}_g(\theta_g; x) \triangleq -\log p(x) = -\sum_{j=1}^{\ell} \log p(x_j; h_{j-1}), \tag{1}$$

where θ_g represents all parameters of RNN. For VRNN, however, the marginal density function $p(x)$ is intractable due to the highly non-linear dependency between x and z. Thus, following Kingma and Welling (2014) and Chung et al. (2015), we consider a variational lower bound of the marginal likelihood function by introducing an approximate posterior model $q(z|x)$, i.e.,

$$\log p(x) \geq E_{z\sim q(z|x)} \log p(x|z) - KL(q(z|x)||p(z)),$$

where KL is the Kullback–Leibler divergence. When the approximate posterior $q(z|x)$ equals the true posterior $p(z|x)$, the gap between the log-likelihood and the lower bound becomes zero. In practice, the approximate posterior model $q(z|x)$ is chosen to be

$$q(z|x) = \prod_{j=1}^{e} q(z_j|x_1,\ldots,x_j, z_1,\ldots,z_{j-1}) = \prod_{j=1}^{e} q(z_j|x_j; h_{j-1}),$$

where $q(z_j|x_j;h_{j-1})$ follows a multivariate Gaussian distribution whose mean and covariance are parameterized by a neural network taking x_j and h_{j-1} as inputs. Overall, the generative model $p(x,z)$ and the approximate posterior model $q(z|x)$ are learned simultaneously by minimizing the negative lower bound

$$\mathcal{L}_g(\theta_g;x) = -E_{z\sim q(z|x)} \sum_{j=1}^{e} \left(\log p(x_j|z_j;h_{j-1}) - KL(q(z_j|x_j;h_{j-1})||p(z_j;h_{j-1}))\right), \tag{2}$$

where θ_g represents all parameters of VRNN.

The supervised loss is given by the cross entropy between the true class label y and the class probabilities returned by the label prediction network

$$\mathcal{L}_d(\theta_d,\tilde{\theta}_g; y,x,w) = -\log p(y|x,w), \tag{3}$$

where θ_d represents all parameters used in the time-static feature extractor ϕ and classifier φ, and $\tilde{\theta}_g$ are the parameters used in h or the approximate posterior model $q(z|x)$ that is a subset of θ_g.

The overall objective function is a weighted sum of the unsupervised loss and the supervised loss

$$\mathcal{L}(\theta_g,\theta_d) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_d(\theta_d,\tilde{\theta}_g;y^i,x^i,w^i) + \eta \cdot \frac{1}{n+m}\sum_{i=1}^{n+m} \mathcal{L}_g(\theta_g;x^i), \tag{4}$$

where the first term is an average over the labelled data, the second term is an average over both labelled and unlabelled data, and η is a weight hyperparameter. The parameters $\tilde{\theta}_g$ are included in both $\mathcal{L}_d$ and $\mathcal{L}_g$ and are iteratively updated using both unlabelled and labelled data during training. Therefore, the representation of longitudinal features is learned not only by the unsupervised generative task but also under the supervision from the labelled data. Further, the hyperparameter η controls the trade-off between the unsupervised learning and the supervised learning. A lower value of η leads to a stronger supervision from labelled data but weaker unsupervised learning from unlabelled data. For example, when η equals to zero, it is equivalent to supervised learning using labelled data alone. Based on our numerical experiments, η is an important hyperparameter that needs to be tuned carefully.

4 | NUMERICAL EXPERIMENTS

To demonstrate the effectiveness of joint learning, we evaluate the proposed method by using five mortality-related classification tasks on the Medical Information Mart for Intensive Care III (MIMIC) database (Johnson et al., 2016; Goldberger et al., 2000). We aim to examine (1) whether additional unlabelled longitudinal features can help improve the prediction performance through semi-supervised learning approaches; (2) how the proposed joint learning method performs in comparison with existing two-step semi-supervised learning methods; and (3) how using the stochastic RNN as the sequence generative model differs from using the deterministic RNN.

4.1 | Datasets

The MIMIC database provides deidentified clinical data of patients admitted to an intensive care unit (ICU) stay. It has been used to benchmark the performance of deep learning models for predicting the length of stay, phenotyping, ICD-9 code group, in-hospital mortality (Harutyunyan, Khachatrian, Kale, Ver Steeg, & Galstyan, 2019), short-term mortality, and long-term mortality (Purushotham, Meng, Che, & Liu, 2018). Nonetheless, the evaluation of semi-supervised learning methods on MIMIC is still lacking. In this paper, we predict five mortality-related tasks: in-hospital mortality, 2-day and 3-day mortality (short-term mortality), and 30-day and 1-year mortality (long-term mortality). We focus on adult patients

TABLE 1 The proportion of in-hospital mortality, 2-day mortality, 3-day mortality, 30-day mortality, and 1-year mortality in the admissions where adult patients were alive the first 24 hours.

Total number of admissions	In-hospital	2-day	3-day	30-day	1-year
35,643	0.105	0.018	0.029	0.147	0.250

who were alive the first 24 hours after the first admission to ICU, which results in an analytic sample of 35,643 patients. Table 1 summarizes the proportion of mortality for each task.

Following Purushotham et al. (2018), we take 15 longitudinal features from the first 24 h after admission to ICU. Specifically, they are the three types of Glasgow Coma Scale scores, systolic blood pressure, heart rate, body temperature, PaO_2, FiO_2, urine output, white blood cell count, serum urea nitrogen level, serum bicarbonate level, sodium level, potassium level, and bilirubin level. Each longitudinal feature is sampled hourly. We also use five time-static features: age, admission type, and three chronic diseases diagnosis including metastatic cancer, hematologic malignancy, and acquired immunodeficiency syndrome.

4.2 | Methods for comparison

Overall, we consider a supervised learning method that uses only labelled data and three semi-supervised learning methods that can use both labelled and unlabelled data. The supervised learning method MMDL combines an RNN for the longitudinal features and a fully connected neural network for the time-static features and is trained by minimizing $\mathcal{L}_d$ in (3) with labelled data only (Purushotham et al., 2018). We use Two-Step to refer to the two-step semi-supervised sequence learning method used in Dai and Le (2015) and Ballinger et al. (2018). Specifically, it shares the same architecture as MMDL, where the RNN is first trained by minimizing $\mathcal{L}_g$ in (1) and then the label prediction network is learned by minimizing $\mathcal{L}_d$ in (3) starting with the pre-trained initialization of $\tilde{\theta}_g$. The proposed methods are referred to as Joint-RNN and Joint-VRNN, and they are trained by minimizing the overall loss $\mathcal{L}$ in (4) jointly. Joint-RNN shares the same architecture with MMDL and Two-Step, while Joint-VRNN substitutes the RNN with a VRNN.

For the comparison to be fair, in MMDL, Two-Step, and Joint-RNN, we adopt exactly the same neural network architecture as used by Purushotham et al. (2018). In Joint-VRNN, we also make the architecture choices as close to the former three models as possible. Specifically, we use GRU for all recurrent units and the sigmoid activation for non-linear transformations except for using a Softmax output layer to return probability scores. Dropout is applied with rate 0.1 after each sigmoid activation in the fully connected neural network. The numbers of layers and hidden units in the recurrent layer h and the fully connected neural networks are the same as those used by Purushotham et al. (2018). For VRNN, we fix the dimension of z_j as 8 and the number of hidden units in the feature extractor $\rho(z_j)$ as 32. Finally, as all patients in this dataset have the same length of longitudinal features, we simply concatenate $h(x)$ or $\tilde{\mu}_z(x)$ over the time steps when sending them as the inputs to the label prediction network, following the implementation of MMDL.

4.3 | Experiment setting

We split the dataset into five folds for stratified cross-validation, among which we use three folds for training, one fold for validation, and the remaining fold for testing. To examine semi-supervised learning methods with various proportions of labelled data, we randomly select a subset of the training folds as labelled data and mask labels of the remaining training folds as unlabelled data. The proportion of labelled training data varies from 1% to 100%. For each classification task, MMDL is learned using only labelled training data, and the other three semi-supervised learning methods are learned using both the labelled and unlabelled training data. We report the mean and standard error of the area under the receiver operating characteristic curve (AuROC) across five testing folds to evaluate the prediction performance.

For the two joint learning methods (Joint-RNN and Joint-VRNN), we grid search the weight hyperparameter η from {0.001,0.01,0.1,1,10} and choose the one with the highest AuROC on the validation fold separately during each round of cross-validation to avoid information leakage. For better pre-training in the Two-Step method, we further tune the non-architecture-specific hyperparameters, including the learning rate and the dropout rate, in the first step. We grid search the optimal learning rate from {0.001,0.005,0.01} and the optimal dropout rate from {0.1,0.2,0.5} with the lowest $\mathcal{L}_g$ on a validation set. In the second step, we initialize $\tilde{\theta}_g$ in the label prediction network with the pre-trained values and train it using labelled data.

All models are implemented in PyTorch and trained with the RMSProp optimizer. We fix the learning rate as 0.001 (except for the pre-training step of Two-Step) and the batch size as 100, following the implementation of MMDL. We use early stopping for all models when reaching the highest AuROC on the validation fold to prevent overfitting.

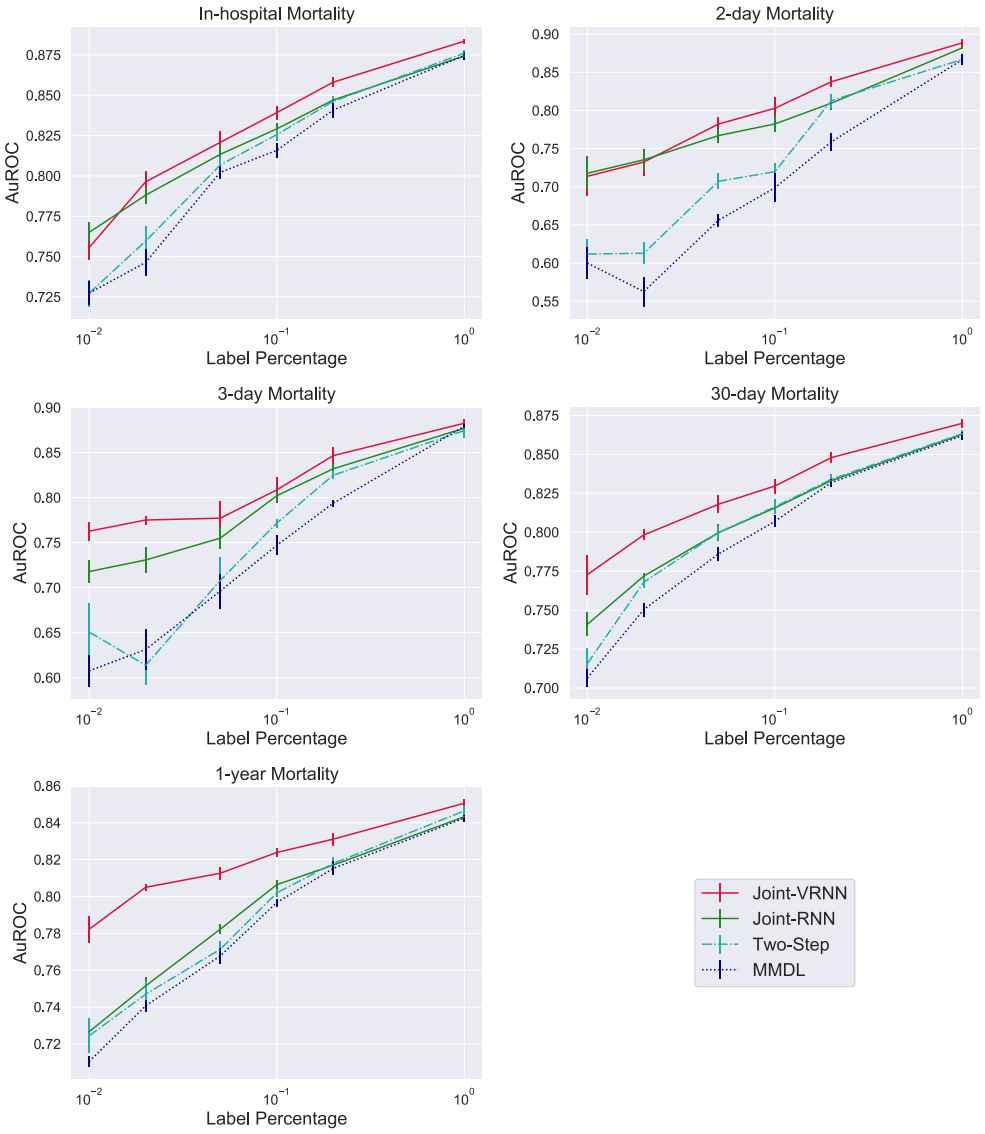

FIGURE 2 Area under the receiver operating characteristic curve (AuROC) of the proposed joint learning methods (Joint-RNN and Joint-VRNN), the two-step method (Two-Step), and the supervised method (MMDL) versus the proportion of labelled training data on five tasks. The horizontal axis is in the logarithmic scale with base 10. The results are averaged over five testing folds, and the error bars indicate the standard error of the mean.

4.4 | Results

Figure 2 shows the AuROC of the four methods under various proportions of labelled training data on five mortality-related classification tasks. First, we observe that when labels are scarce, semi-supervised learning methods significantly outperform the supervised method (MMDL) on the five tasks in most cases. This implies that semi-supervised learning methods which incorporate unlabelled data can help improve prediction performance compared to the supervised method which uses labelled data only. Further, we notice that, even in the fully labelled case, i.e. when the label percentage is 100%, modelling what comes next in a sequence as an auxiliary task (as joint learning methods do) could further improve the performance on classification tasks.

Second, we observe that the joint learning methods obtain a higher AuROC by a large margin compared to the existing two-step method, especially when predicting short-term mortality. Moreover, the gain of the joint learning methods increases as the label percentage decreases. This implies that, although the pre-training step of the two-step method might provide a potentially good initialization, the lack of supervision from labels in the pre-training step would lead to limited improvement on prediction performance in the second step. Instead, the proposed joint learning methods can take advantage of available labels and learn representations of the longitudinal features under supervision from both labelled and unlabelled data.

Third, as shown in Figure 2, the Joint-VRNN that contains the stochastic recurrent layer further improves the prediction performance in comparison with the Joint-RNN. The gain is especially obvious for the long-term mortality prediction. This extends the observation of the benefit of using latent random recurrent layers in previous literature to modelling longitudinal features.

5 | DISCUSSION

In this paper, we propose a semi-supervised joint learning method for classifying longitudinal features, with an application to clinical events. With joint learning, the feature representation of the longitudinal information is learned under supervision from both unlabelled and labelled data so that related data can be separated well corresponding to the labels. We compare the proposed methods with the existing supervised learning method and two-step semi-supervised learning method. Our experimental results verify that, by incorporating unlabelled data, semi-supervised learning methods outperform the supervised method when labels are scarce, and among the semi-supervised learning methods, the proposed joint learning methods can further improve the prediction accuracy compared to the two-step method in most cases.

Notably, the horizontal difference between the curves of semi-supervised and supervised methods indicates the difference on the usage of labelled training data to maintain the same prediction performance. For example, as shown in Figure 2, the Joint-VRNN method uses 2% labels to obtain 80% AuROC for 1-year mortality prediction while the supervised method needs 10% labels to achieve the same performance. Therefore, the usage of semi-supervised learning methods could help reduce the cost of collecting clinical labels when building prediction models for applications in health care.

We should note a few remarks on the proposed joint learning methods. First, when there are multiple prediction tasks, the two-step method has the advantage that the pre-training step only needs to be done once while the joint learning methods require the training of the sequence generative network for each task. Second, compared to the two-step method, the joint learning methods have one additional hyperparameter η to be tuned. In practice, though, a simple grid search of η is enough to obtain good performance as shown in our experiments. Third, Joint-VRNN has a higher computational cost than Joint-RNN due to the sequential sampling. However, Joint-VRNN demonstrates promising improvements in prediction accuracy compared to Joint-RNN, which is especially important in health care applications.

ACKNOWLEDGEMENT

We would like to thank Chenkai Sun for his help on part of the baseline experiments. This research was partially supported by National Science Foundation grant DMS-1821243.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Ji Zhu https://orcid.org/0000-0002-7812-5378

REFERENCES

Ballinger, B., Hsieh, J., Singh, A., Sohoni, N., Wang, J., Tison, G. H., … Pletcher, M. J. (2018). DeepHeart: Semi-supervised sequence learning for cardiovascular risk prediction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, pp. 2079-2086.

Che, Z., Cheng, Y., Zhai, S., Sun, Z., & Liu, Y. (2017). Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In *2017 IEEE International Conference on Data Mining (ICDM)*, New Orleans, LA, pp. 787–792. https://doi.org/10.1109/ICDM.2017.93

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179

Choi, E., Schuetz, A., Stewart, W. F., & Sun, J. (2016). Using recurrent neural network models for early detection of heart failure onset. Journal of the American Medical Informatics Association, 24(2), 361–370. https://doi.org/10.1093/jamia/ocw112

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., & Bengio, Y. (2015). A recurrent latent variable model for sequential data, *In Advances in Neural Information Processing Systems* (pp. 2980–2988). Montreal, Canada: Curran Associates, Inc.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning, *In Advances in Neural Information Processing Systems* (pp. 3079–3087). Montreal, Canada: Curran Associates, Inc.

Esteban, C., Staeck, O., Baier, S., Yang, Y., & Tresp, V. (2016). Predicting clinical events by combining static and dynamic information using recurrent neural networks. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, Chicago, IL, pp. 93–101. https://doi.org/10.1109/ICHI.2016.16

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., & Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. Circulation, 101(23), e215–e220. https://doi.org/10.1161/01.CIR.101.23.e215

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets, *In Advances in Neural Information Processing Systems* (pp. 2672–2680). Montreal, Canada: Curran Associates, Inc.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., & Webster, D. R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Journal of the American Medical Association (JAMA), 316(22), 2402–2410. https://doi.org/10.1001/jama.2016.17216

Harutyunyan, H., Khachatrian, H., Kale, D. C., VerSteeg, G., & Galstyan, A. (2019). Multitask learning and benchmarking with clinical time series data. Scientific Data, 6(1), 1–18. https://doi.org/10.1038/s41597-019-0103-9

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. Scientific Data, 3(1), 1–9. https://doi.org/10.1038/sdata.2016.35

Kaji, D. A., Zech, J. R., Kim, J. S., Cho, S. K., Dangayach, N. S., Costa, A. B., & Oermann, E. K.(2019). An attention based deep learning model of clinical events in the intensive care unit. PLOS One, 14(2), e0211057. https://doi.org/10.1371/journal.pone.0211057

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models, *In Advances in neural information processing systems* (pp. 3581–3589). Montreal, Canada: Curran Associates, Inc.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations (ICLR)*, Banff, Canada.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Narayanaswamy, S., Paige, T. B., Vande Meent, J. W., Desmaison, A., Goodman, N., Kohli, P., & Torr, P. (2017). Learning disentangled representations with semi-supervised deep generative models, *In Advances in neural information processing systems* (pp. 5925–5935). Long Beach, CA: Curran Associates, Inc.

Odena, A. (2016). Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583.

Purushotham, S., Meng, C., Che, Z., & Liu, Y. (2018). Benchmarking deep learning models on large healthcare datasets. Journal of Biomedical Informatics, 83, 112–134. https://doi.org/10.1016/j.jbi.2018.04.007

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, WA, pp. 1631–1642.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., … Klingner, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Yao, L., Mao, C., & Luo, Y. (2019). Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. BMC Medical Informatics and Decision Making, 19(Suppl 3), 71. https://doi.org/10.1186/s12911-019-0781-4

How to cite this article: Tang W, Ma J, Waljee AK, Zhu J. Semi-supervised joint learning for longitudinal clinical events classification using neural network models. *Stat*. 2020;9:e305. https://doi.org/10.1002/sta4.305