Improving Lives of Indebted Farmers Using Deep Learning: Predicting Agricultural Produce Prices Using Convolutional Neural Networks

Hangzhi Guo,[1] **Alexander Woodruff,**[2] **Amulya Yadav**[2]
[1]Wenzhou Kean University, Wenzhou, China 325060
[2]Pennsylvania State University, University Park, PA 16802
guoha@kean.edu, {arw5550, amulya}@psu.edu

Abstract

Farmer suicides have become an urgent social problem which governments around the world are trying hard to solve. Most farmers are driven to suicide due to an inability to sell their produce at desired profit levels, which is caused by the widespread uncertainty/fluctuation in produce prices resulting from varying market conditions. To prevent farmer suicides, this paper takes the first step towards resolving the issue of produce price uncertainty by presenting PECAD, a deep learning algorithm for accurate prediction of future produce prices based on past pricing and volume patterns. While previous work presents machine learning algorithms for prediction of produce prices, they suffer from two limitations: (i) they do not explicitly consider the spatio-temporal dependence of future prices on past data; and as a result, (ii) they rely on classical ML prediction models which often perform poorly when applied to spatio-temporal datasets. PECAD addresses these limitations via three major contributions: (i) we gather real-world daily price and (produced) volume data of different crops over a period of 11 years from an official Indian government administered website; (ii) we pre-process this raw dataset via state-of-the-art imputation techniques to account for missing data entries; and (iii) PECAD proposes a novel wide and deep neural network architecture which consists of two separate convolutional neural network models (trained for pricing and volume data respectively). Our simulation results show that PECAD outperforms existing state-of-the-art baseline methods by achieving significantly lesser root mean squared error (RMSE) - PECAD achieves $\sim$25% lesser coefficient of variance than state-of-the-art baselines. Our work is done in collaboration with a non-profit agency that works on preventing farmer suicides in the Indian state of Jharkhand, and PECAD is currently being reviewed by them for potential deployment.

Introduction

In the last two decades, the issue of agrarian distress (and other related socio-economic problems such as indebtedness, loss of agricultural income, etc.) have led to a significant increase in suicide rates among small-scale farmers, especially in developing countries such as India, Pakistan, etc. Around 300,000 Indian farmers have committed suicide since 1995. As of 2014, 60,000 farmers committed suicide in the Indian state of Maharashtra alone, with an average of 10 suicides every day (NCRB 2019).

There are a myriad of factors that lead to farmer suicides, e.g., crop failures, low farm productivity, an inability to achieve profits, inefficient cold chain management resulting in wastage of agricultural produce, lack of irrigation facilities, and insurmountable debt. However, one key factor that contributes to farmer suicides is the uncertainty associated with agricultural prices and markets, i.e., variations in global market conditions can lead to abrupt fluctuations in prices of agricultural produce at a local level (Barik 2018). Due to this uncertainty over prices, indebted small-scale farmers who often lack advanced technological resources and knowledge about global market conditions are unable to make accurate decisions about when (and where) to sell their produce. As a result, they are unable to earn desired profits on their produce and repay their agricultural loans (see Figures 1a and 1b), which causes many of these farmers to commit suicide (Panagariya 2008).

(a) Farmers Protesting by Throwing their Unsold Produce

(b) Huge Demand for Loan Waiver at Farmer Rally

Figure 1: Agrarian Distress in India

Thus, immediate steps need to be taken to alleviate issues of these farmers. Recent advances in Machine Learning (ML) techniques have made it possible to apply learning algorithms successfully to different social problems (Tambe and Rice 2018). As a first step in solving problems of farmers outlined above, this paper proposes an AI/ML approach to answer the following question: *Can data-driven approaches use historical pricing and volume patterns at different markets to predict future prices of agricultural produce at these markets?* These AI/ML approaches can then

be used by farmers to select intelligent strategies for selling their produce, e.g., via future price predictions, farmers can decide when (in the future) they should sell their produce in order to maximize their profit.

There are several challenges that need to be solved to answer this question. First, existing datasets on pricing patterns[1] are very sparse (i.e., they have lots of missing entries), which hinders the training process. Second, future produce prices have a long-term temporal dependence on past prices (e.g., the price of tomatoes in August 2019 may depend on their price in August 2018) and a spatial dependence on the prices at nearby markets (e.g., prices at nearby markets may be similar, as opposed to geographically distant markets), and thus, it is important to develop prediction models which can explicitly capture this spatio-temporal dependence.

While previous work presents algorithms to predict future produce prices, they (i) do not explicitly consider the spatio-temporal dependence of future prices on past data; and as a result, (ii) they rely on classical ML prediction models (e.g., decision trees) which often perform poorly when applied to spatio-temporal datasets (we validate this in our experiments). These shortcomings limit the accuracy (and hence, usability) of these methods in the real-world.

In this paper, we address these shortcomings by proposing PECAD (**P**rice **E**stimation for **C**rops using the **A**pplication of **D**eep Learning), a novel neural network architecture to predict future prices of agricultural produce. In order to address the shortcomings in previous work, PECAD makes the following novel contributions. First, it collects real-world prices and (produced) volume of different crops at ∼1,350 agricultural markets in India over a period of 11 years (i.e., 2008 to 2018) from Agmarknet.gov.in[1] (an official Indian government administered website). Second, PECAD preprocesses this raw dataset via state-of-the-art imputation (and other) techniques to account for missing data entries. Third, using this data as input, PECAD proposes a novel neural network architecture inspired by the wide and deep learning paradigm (Cheng et al. 2016), which jointly trains wide linear models and deep neural networks. However, instead of using cross-product feature transformations as input to the wide linear models, PECAD uses a novel combination of two separate convolutional neural network (CNN) models for pricing and volume data respectively (for the crop under consideration), and uses these CNN models as input to the wide linear model. Our simulation results show that PECAD significantly outperforms existing state-of-the-art baseline methods – it achieves 25% lesser coefficient of variance than baseline methods, which emphasizes the importance of explicitly modeling the spatio-temporal dependence of future prices on past data inside our ML algorithm. Our work is done in collaboration with a non-profit agency that works on preventing farmer suicides in the Indian state of Jharkhand (name withheld for anonymity), and PECAD is currently being reviewed by them for potential deployment.

Related Work We discuss prior AI/ML research that assists in alleviating agrarian distress. (You et al. 2017) proposed deep Gaussian processes to predict crop yields using remote sensing data. However, their approach relies on gathering satellite images of fields, which can be expensive to obtain in low-resource environments in developing countries. In our work, we rely on easily available pricing and volume data to predict future prices. Next, (Chen, Nowocin, and Marathe 2017) proposed a hardware and software solution to reduce spoilage of agricultural crops. (Ma et al. 2019) is most closely related to our research, as they also build a crop price prediction model using data from the same source[1]. Unfortunately, they fail to exploit spatio-temporal properties of pricing and (produced) volume data for different crops, which leads to poor performance accuracy (as we show in our experiments). In our work, we use specific forms of convolutional neural networks to uncover spatio-temporal dependencies in pricing and volume data.

[1] http://agmarknet.gov.in/

Dataset Construction

Data Collection We rely on two different data sources. We collect all our raw data on agricultural crops (produce) from Agmarknet.gov.in[1], a website run by the Indian government's Ministry of Agriculture and Farmers Welfare, which contains daily price and volume data at 1352 agricultural markets across India for over twelve years. For our paper, we collected price and volume data for three different crops (Brinjal, Tomato, and Chili) across all markets for a period of 11 years (2008 to 2018). To effectively retrieve this data, we deployed a multi-process crawler script on two cloud servers to scrape the market data from this website. This entire scraping process took one week to complete.

In addition, we augment this data by collecting spatial features for each agricultural market, e.g., the geographical location of the market. We collect this data to capture the spatial correlations between crop prices at geographically close markets (i.e., crop prices at markets situated close to each other are likely to be similar). Since Agmarknet.gov.in does not contain any spatial information about the 1352 markets in its database, we use Google Maps API to obtain the geometric coordinate information (latitude and longitude) for each market in the Agmarknet.gov.in database. Moreover, we assign each market and crop with a unique ID, and we represent this feature with sparse one-hot encoding vectors.

Data Preprocessing Let $\boldsymbol{M}$ denote the set of all 1352 markets in our dataset, $\boldsymbol{C}$ denote the set of produce types (we collect data for three crops, so $|\boldsymbol{C}| = 3$), and $\boldsymbol{T}$ denote the set of all dates (timesteps) for which we have price and volume entries. For each crop $c \in \boldsymbol{C}$, we define P^c and V^c as $\boldsymbol{M} \times \boldsymbol{T}$ price and volume matrices (respectively). For each $m \in \boldsymbol{M}$ and $t \in \boldsymbol{T}$, $P^c_{m,t}$ indicates the price of crop c in market m on day t, whereas $V^c_{m,t}$ indicates the volume of crop c (in metric tonnes) that arrived in market m on day t.

Unfortunately, the P^c and V^c matrices for each crop $c \in \boldsymbol{C}$ (which we construct after data collection) are extremely sparse, i.e., they have several missing entries. On Agmarknet.gov.in, these missing entries are created due to a variety of reasons, e.g., a particular market might have been closed on a given day $t \in \boldsymbol{T}$, no produce was sold in a market on a given day, or simply the data for that market was never recorded due to human errors. In particular, we ob-

Feature	Explanation	Notation
Market	Unique identifier for each market	$m \in \boldsymbol{M}$
Crop	Unique identifier for each crop	$c \in \boldsymbol{C}$
Price	Denotes the price of crop c in market m on day t	$P^c_{m,t}$
Volume	Denotes the volume of crop c in market m on day t	$V^c_{m,t}$
Geo location	Denotes the geographical latitude/longitude of market m	$[lat_m, lon_m]$

Table 1: Features and notations in our paper.

serve that the data for some markets is extremely sparse, i.e., there exist very few valid (non-empty) data points for some markets and data from these markets has little significance in the overall learning process. Therefore, we eliminate the data from all those markets which have valid data entries for less than 10% of days in $\boldsymbol{T}$.

Next, we use effective data imputation methods to extrapolate remaining missing values. Given the sparsity of the crop pricing/volume data, naive imputation methods (e.g., hot-deck, mean substitution) are not applicable. However, given the spatial correlations between crop prices (and volumes) at geographically close markets, we use SoftImpute (Hastie et al. 2015), a state-of-the-art collaborative filtering method to extrapolate missing values in our dataset.

After data imputation, we have completely filled P^c and V^c matrices for each crop $c \in \boldsymbol{C}$. Unfortunately, since most sequential neural networks suffer from vanishing (or exploding) gradients (which results in an inability to learn long-term temporal dependencies) (Sutskever, Vinyals, and Le 2014), we compress P^c and V^c matrices by considering a time window of w days as a single time step. Formally, for every non-overlapping consecutive block of w days, we average the crop prices and volumes to obtain compressed price and volume matrices $\hat{P}^c_{m,t} = \frac{1}{w} \sum_{tw}^{(t+1)w-1} P^c_{m,t}$ and $\hat{V}^c_{m,t} = \frac{1}{w} \sum_{tw}^{(t+1)w-1} V^c_{m,t}$. Note that if w does not divide $|\boldsymbol{T}|$, we ignore the last time window of length $l < w$.

Data Characteristics Our final dataset has ∼40000 datapoints, each consisting of a feature vector and a continuous label. A single feature vector for the t^{th} time-step at market m consists of historical price and volume pairs for the last n time-steps from our compressed $\hat{P}^c$ and $\hat{V}^c$ matrices, along with market latitude/longitude coordinates, and market and crop identifiers. The ground-truth label (which we want to predict) is the price of crop c at market m on the $(t+1)^{th}$ time-step (i.e., the crop price in the next time-step). Table 1 describes a list of all features in our dataset.

Deep Learning Algorithm

We now describe PECAD, our novel deep learning architecture which is inspired by wide and deep networks (Cheng et al. 2016). For completeness, we first provide a short description of wide and deep networks, and temporal convolutional networks (TCN) (Bai, Kolter, and Koltun 2018), which form building blocks of our PECAD architecture.

Wide and Deep Networks The wide and deep network model consists of jointly trained wide linear models and deep neural networks (see Figure 2), and this model is highly effective for large-scale regression problems with sparse inputs, i.e., categorical features with a large number of possible feature values (Cheng et al. 2016). This makes the wide and deep learning paradigm an ideal fit for PECAD, as our price prediction dataset contains highly sparse one-hot encoding feature vectors to identify markets and crops.

The deep component is a feed-forward neural network, as shown in Figure 2 (right). The first layer of the deep component converts high-dimensional and sparse one-hot encoding vectors into low-dimensional and dense real-valued vectors, often referred to as embedding vectors. These dense embedding vectors (see right side of Figure 2) are then fed into the hidden layers of the neural network.

The wide component is a generalized linear model (GLM) of the form $y = \boldsymbol{w}^T\boldsymbol{x} + b$, as illustrated in Figure 2 (left). Let y denote the prediction, $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$ is a vector of d features, $\boldsymbol{w} = [w_1, w_2, \ldots, w_d]$ are the model parameters and b is the bias. Importantly, the feature vector $\boldsymbol{x}$ for the wide linear model includes cross-product transformation features, which capture interactions between the input binary features, and adds non-linearity to the GLM.

The outputs from the wide and deep components are combined using a weighted sum of their output log odds as the prediction (see middle part of Figure 2), which is then fed to one common logistic loss function for joint training.

Temporal Convolutional Networks The TCN model (Bai, Kolter, and Koltun 2018) utilizes convolutional layers to deal with sequential information. TCNs can take a sequence of any length and map it to an output sequence of the same length, similar to standard RNN models. To accomplish this, the TCN model uses a 1D fully-convolutional network (FCN) architecture, where each hidden layer is the same length as the input layer, and zero padding of length $(\text{filter size} - 1)$ is added to keep subsequent layers the same length as previous ones. In addition, TCN uses causal convolutions (in which an output at time t is convolved only with elements from time t and earlier in the previous layer) to ensure that there is no information leakage from future to past. Finally, to extract correlations in long-term sequences, TCN utilizes dilated convolutions (which leads to an exponential increase in the receptive field of convolutional filters). TCN has been demonstrated to outperform state-of-the-art recurrent architectures such as LSTM on diverse benchmarks and tasks. Given the superior properties of TCN, we utilize this structure as a building block inside PECAD.

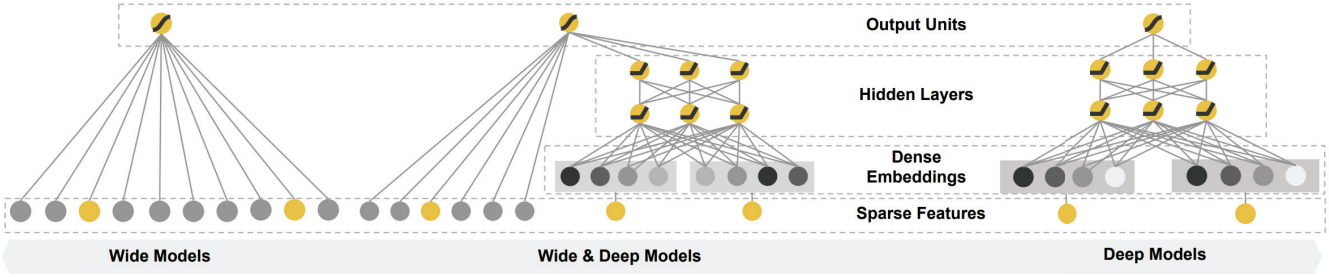

Figure 2: Wide and Deep Network Architectures (Cheng et al. 2016)

PECAD: Deep Learning Architecture

We now describe the wide and deep learning architecture of PECAD. The wide linear models in PECAD are used to memorize long-term sequential pricing/volume information, whereas the deep neural networks in PECAD are used to generalize to previously unseen feature interactions through low-dimensional embeddings. Note that in standard wide and deep networks (Cheng et al. 2016), the feature vector $\boldsymbol{x}$ for the wide component includes cross-product transformation features. Unfortunately, in the produce price prediction problem, the number of these features grow exponentially with the length of price/volume history under consideration, which hinders the learning performance of standard wide and deep networks. A key novelty inside PECAD is that instead of adding exponentially many cross-product transformation features, PECAD adds non-linearity to the wide component (GLM) by training two separate TCN models for price and volume, respectively. We validate the benefit of adding the TCN models as input to the wide component (GLM) in our experiments.

First, we describe a high-level overview of PECAD's entire architecture. Next, we describe the details of PECAD's embedding layer and its wide and deep networks.

Architecture Overview Figure 3 shows the entire architecture of PECAD. The left and right part of Figure 3 corresponds to the wide network (GLM) and deep network, respectively. The feature set of the wide network GLM does not consist of the raw input features; instead, we train separate TCN models which take as input raw historical price and volume patterns (respectively), and produce complicated non-linear features which form the feature vectors for the wide network. On the other hand, the feature vector of the deep network includes embedding vectors for markets and crops, spatial features of agricultural markets (e.g., geographical latitude and longitude coordinates), in addition to the complicated non-linear feature vectors that were fed into the wide network (see Figure 3). Finally, the output from the wide and deep networks is combined and fed into a single fully connected layer, which outputs a prediction of the produce price on the next day.

Embedding Layer We assign unique identifiers for each of the $|\boldsymbol{M}| = 1352$ markets and $|\boldsymbol{C}| = 3$ crops, and represent this feature using extremely sparse one-hot encoding feature vectors, which leads to poor learning performance. Thus, we apply an embedding layer to transform sparse high-dimensional datapoints (which contain one-hot encoding vectors for the market and crop) into low-dimensional embedding vectors: $\mathbf{v}_{\mathrm{e},i} = W \cdot \mathbf{v}_i$, where $\mathbf{v}_{\mathrm{e},i} \in \mathbb{R}^{d_e}$ is the embedding vector for the i^{th} datapoint, $\mathbf{v}_i \in \mathbb{R}^{d_x}$ is the i^{th} datapoint with one-hot encoding ($d_e < d_x$). The embedding parameter matrix $W \in \mathbb{R}^{d_e \times d_x}$ is initialized randomly, and is updated during model training to minimize loss.

PECAD Deep Network The deep network is a feed-forward deep neural network (DNN) which takes the low-dimensional embedding vectors as input. The DNN contains three rectified linear unit (ReLU) fully connected hidden layers, which can be denoted as: $\mathbf{h}_{l+1} = \mathrm{ReLU}(W_l \mathbf{h}_l + \mathbf{b}_l)$, where $\mathbf{h}_{l+1}$, $\mathbf{h}_l$ are the $(l+1)$-th and l-th hidden layer, respectively; $W_l \in \mathbb{R}^{d_{l+1} \times d_l}$, $b_l \in \mathbb{R}^{d_{l+1} \times d_l}$ are weights and bias for the l-th fully-connected layer, respectively.

PECAD Wide Network The wide network consists of two TCNs trained separately on sequential price and volume data (respectively), and their output is fed into a GLM. Conventionally, multivariate time-series data are stacked into a single TCN network, i.e., a single TCN model could be trained for both price and volume time-series. However, PECAD trains two independent TCN models for memorizing long-term sequential price and volume information (respectively). In our experiments, we empirically validate the choice of using two separate TCN models inside PECAD by comparing its predictive performance against a variant of PECAD which trains a single stacked TCN model (referred to as *PECAD-Single TCN* in Table 2).

Training Procedure We temporally split the data into a training/test set. Pricing and volume data from 2008 to 2016 (along with other features in Table 1) is used as the training data. We train PECAD on this training data and use data from 2017-2018 as the test set. Finally, we employ an L_2 loss function, i.e., $L_2 = \sum_{i=1}^{n} (\hat{y} - y_{predicted})^2$.

Experimental Evaluation

All experiments were run on an Ubuntu based Deep Learning Amazon Machine Image (AMI) Version 24.0 server. In all experiments, we use pricing and volume data for the previous $n = 360$ days as part of the feature space in our dataset. All deep learning models (PECAD and other baselines) are trained for 150 epochs, and their performance is

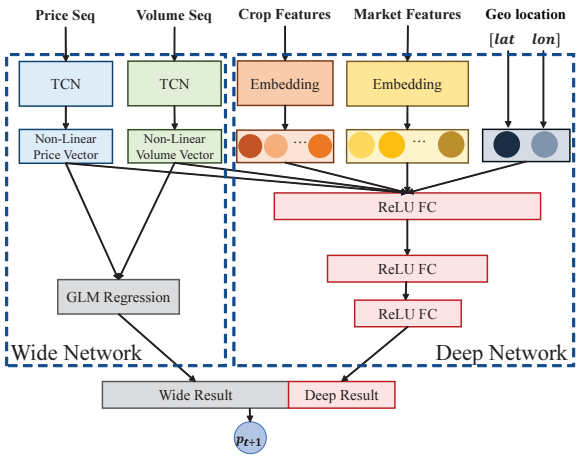

Figure 3: Architecture of our deep learning model

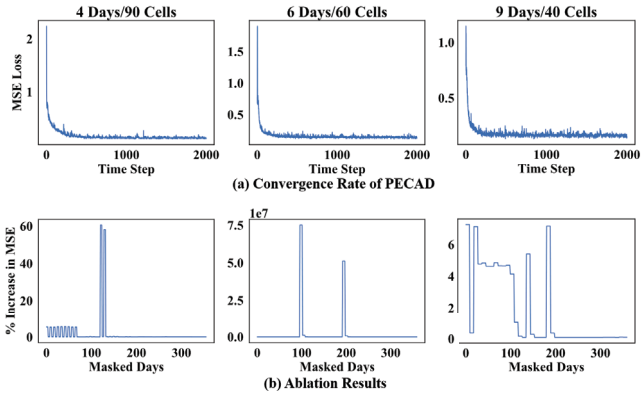

Figure 4: Convergence Rate & Ablation Results for PECAD

averaged over 20 runs. To compare predictive performance of different algorithms, we use "*coefficient of variation*" (i.e., the root mean squared error (RMSE) divided by the mean produce price) as our comparison metric, instead of RMSE values (which cannot be compared across different models in a meaningful way to determine which model provides better predictions of an outcome) (Sørensen 2002).

Baselines We compare against two classical ML models: (i) random forests (*RF*); and (ii) gradient tree boosting (*GTB*). We use these two baselines as they are the best performing algorithms for produce price prediction (Ma et al. 2019). In addition, we also compare against four deep learning models, which also utilize spatio-temporal features: (i) standard TCN model (*TCN*); (ii) LSTM networks with an attention layer (*Attention-LSTM*) (Sutskever, Vinyals, and Le 2014); (iii) standard wide and deep networks (*Standard Wide & Deep*) with cross-product transformation features (Cheng et al. 2016); and (iv) PECAD with a single TCN for both price and volume sequences (*PECAD-Single TCN*).

Predictive Performance Table 2 compares the *coefficient of variation* achieved by different ML models for three crops (Brinjal, Tomato and Chilli) across three different time window sizes ($w = 4, 6$ and 9 days). The results in Table 2 illustrate the benefit of explicitly considering the spatio-temporal dependence of future produce prices on past data - our deep learning models which explicitly consider spatio-temporal dependence perform significantly better in predicting produce prices accurately (as compared to classical ML algorithms). Specifically, for each w value and for each crop, we observe that deep learning models have the lowest *coefficient of variation* (highlighted in bold) in each case. In particular, deep learning models achieve 12.5% lesser *coefficient of variation* as compared to the two classical ML models.

Further, Table 2 shows that PECAD significantly outperforms other deep learning models by achieving ∼25% lesser *coefficient of variation* as compared to the average case performance of the other four deep learning models. In particular, PECAD outperforms the standard deep and wide network model (Cheng et al. 2016) by achieving ∼13% lesser coefficient of variation, which illustrates the shortcomings of using standard cross-product transformation features in the price prediction problem. Further, PECAD outpeforms *PECAD-Single TCN* by ∼13.5%, which illustrates the benefit of training two separate TCN models in the PECAD architecture. This figure establishes PECAD's superior performance in predicting future produce prices.

Convergence Results Figure 4a analyzes PECAD's rate of convergence for different time window sizes (averaged across all three crops). The X-axis shows increasing time epochs, and the Y-axis shows the mean squared error (MSE) training loss. This figure shows that PECAD converges fairly quickly to locally optimal solutions.

Ablation Studies We explore the impact of various parts of our feature space on PECAD's predictive accuracy. Thus, we experiment with various ablations of our PECAD model obtained by excluding key components from the feature space one at a time. We generate ablated models by masking price and volume entries for each of the last $n = 360$ days. Figure 4b shows the effect of ablating different parts of the feature space. The X-axis shows the day for which price/volume entries are masked to get an ablated model. The Y-axis shows the average percentage increase in MSE loss as a result of the ablation (averaged across all three crops). For example, if price and volume entries for the last (most recent) day are masked (X-axis label = 1), the MSE loss increases by 5% in the PECAD model trained with a time window size of $w = 4$ days. Surprisingly, Figure 4b shows that across all three time window sizes, price/volume entries around days 100 to 150 (in the past) are important in predicting future produce prices, as the MSE loss increases significantly when price/volume entries of days in this time period are masked. One hypothesis to explain this result is that all our crops (Brinjal, Tomato and Chilli) have fixed sowing periods (every year) and have an average growing period of ∼3-4 months (DARD 2019), hence fresh supplies of produce enters markets after every 3-4 months (90-120 days). Thus, price/volume entries corresponding to fresh produce supplies that were recorded 3-4 months ago potentially serve as important predictors for next-day crop prices.

	4 Days/90 Cells			6 Days/60 Cells			9 Days/40 Cells		
	Brinjal	**Tomato**	**Chilli**	**Brinjal**	**Tomato**	**Chilli**	**Brinjal**	**Tomato**	**Chilli**
RF	21.12	22.88	19.45	23.47	38.48	21.60	24.50	44.30	23.54
GTB	21.38	20.85	18.99	22.88	26.18	18.58	23.64	31.55	20.36
Attention-LSTM	19.88	20.52	17.49	21.98	24.36	18.44	**21.00**	31.94	21.04
TCN	20.59	**19.87**	17.36	54.42	33.25	27.69	27.59	98.02	81.83
Standard Wide & Deep	23.63	24.47	19.07	24.34	28.22	18.67	27.36	34.29	21.10
PECAD - Single TCN	21.90	23.50	17.77	29.43	30.86	20.21	26.26	33.65	20.46
PECAD	**19.64**	21.62	**17.07**	**21.14**	**24.20**	**17.65**	21.75	**28.46**	**19.31**

Table 2: Coefficient of Variation of different ML models with varying time window sizes

Implementation Challenges & Conclusion

A few implementation challenges need to be solved when PECAD gets deployed by non-profit agencies working with indebted farmers. First, PECAD's predictive performance can potentially be improved by incorporating historical weather patterns, which can play a role in determining future crop supply (and hence, the future crop price). However, deep learning methods are rarely used to model weather in the real-world, as physical models are far more accurate at predicting future weather. Thus, PECAD needs to be integrated with physical weather prediction models (as part of future work). Further, sophisticated deep learning approaches to predicting future produce prices (such as PECAD) may raise suspicions among low-literate farmers. Public awareness campaigns in the agencies working with this program would help overcome such fears and to encourage participation. Also, non-profit agencies often do not prioritize spending their limited resources to buy sophisticated computer hardware (to train and run PECAD). Thus, we propose deploying PECAD as a stand-alone web service that the agencies could use without our intervention. Finally, PECAD represents a single piece of the puzzle that needs to be solved for preventing farmer suicides, there are many other pieces. For example, PECAD's successful deployment depends crucially on availability of long-term crop pricing and volume patterns. While Agmarknet.gov.in makes this information available for Indian markets, there are no analogous data repositories for other developing countries.

This paper presents PECAD, a deep learning algorithm for accurate prediction of future produce prices based on past pricing and volume patterns. Previous ML algorithms for predicting produce prices do not explicitly consider the spatio-temporal dependence of future prices on past data, which leads to significant shortcomings. PECAD handles these issues by proposing a novel wide and deep learning architecture in which two separate convolutional neural network models are trained for pricing and volume data respectively (for the crop under consideration). Our simulation results show that PECAD outperforms existing state-of-the-art baseline methods by achieving 25% lesser *coefficient of variation*. Our work is done in collaboration with an Indian non-profit agency in the Indian state of Jharkhand that works on preventing farmer suicides, and PECAD is currently being reviewed by them for potential deployment.

References

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Barik, N. 2018. Analysis of interventions addressing farmer distress in rajasthan. https://www.copenhagenconsensus.com/sites/default/files/raj_farmer_distress_sm.pdf.

Chen, G. H.; Nowocin, K.; and Marathe, N. 2017. Toward reducing crop spoilage and increasing small farmer profits in india: a simultaneous hardware and software solution. In *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development*.

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, 7–10. ACM.

DARD. 2019. Vegetable Production in Kwazulu-Natal: Length of Growing Period. https://www.kzndard.gov.za/images/Documents/Horticulture/Veg_prod/length_of_growing_period.pdf.

Hastie, T.; Mazumder, R.; Lee, J. D.; and Zadeh, R. 2015. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research* 16(1):3367–3402.

Ma, W.; Nowocin, K.; Marathe, N.; and Chen, G. H. 2019. An interpretable produce price forecasting system for small and marginal farmers in india using collaborative filtering and adaptive nearest neighbors. In *Proceedings of the Tenth International Conference on Information and Communication Technologies and Development*, 6. ACM.

NCRB. 2019. National Crime Records Bureau. http://ncrb.gov.in/.

Panagariya, A. 2008. *India: The emerging giant*. Oxford University Press.

Sørensen, J. B. 2002. The use and misuse of the coefficient of variation in organizational demography research. *Sociological methods & research* 30(4):475–491.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Tambe, M., and Rice, E. 2018. *Artificial Intelligence and Social Work*. Artificial Intelligence for Social Good. Cambridge University Press.

You, J.; Li, X.; Low, M.; Lobell, D.; and Ermon, S. 2017. Deep gaussian process for crop yield prediction based on remote sensing data. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 4559–4565. AAAI Press.