A Dynamically Adaptive Movie Occupancy Forecasting System with Feature Optimization

Sundararaman Venkataramani
Computer Science
SSN College of Engineering
sundararaman15129@cse.ssn.edu.in

Ateendra Ramesh
Computer Science and Engineering
University at Buffalo
ateendra@buffalo.edu

Sharan Sundar S
Computer Science
SSN College of Engineering
sharansundar15096@cse.ssn.edu.in

Aashish Kumar Jain
Computer Science and Engineering
University at Buffalo
ajain28@buffalo.edu

Gautham Krishna Gudur
Global AI Accelerator
Ericsson
gautham.krishna.gudur@ericsson.com

Vineeth Vijayaraghavan
Director - Research and Outreach
Solarillion Foundation
vineethv@ieee.org

***Abstract*—Demand Forecasting is a primary revenue management strategy in any business model, particularly in the highly volatile entertainment/movie industry wherein, inaccurate forecasting may lead to loss in revenue, improper workforce allocation and food wastage or shortage. Predominant challenges in Occupancy Forecasting might involve complexities in modeling external factors – particularly in Indian multiplexes with multilingual movies, high degrees of uncertainty in crowd-behavior, seasonality drifts, influence of socio-economic events and weather conditions. In this paper, we investigate the problem of movie occupancy forecasting, a significant step in the decision-making process of movie scheduling and resource management, by leveraging the historical transactions performed in a multiplex consisting of eight screens with an average footfall of over 5500 on holidays and over 3500 on nonholidays every day. To effectively capture crowd behavior and predict the occupancy, we engineer and benchmark behavioral features by structuring recent historical transaction data spanning over five years from one of the top Indian movie multiplex chains, and propose various deep learning and conventional machine learning models. We also propose and optimize on a novel feature called *Sale Velocity* to incorporate the dynamic crowd behavior in movies. The performance of these models are benchmarked in real-time using *Mean Absolute Percentage Error (MAPE)*, and found to be highly promising while substantially outperforming a domain expert's predictions.**

***Keywords*—Movie Occupancy Forecasting, Feature Engineering, Machine Learning, Predictive Modeling, Time-Series Forecasting**

I. Introduction

In commercial cinema industry, occupancy prediction has prime importance in organizing and decision making. Foretelling demand allows management personnel to plan appropriately on issues as workforce allocation, supplies, financial budgeting, pricing and inventory. According to the Ficci-KPMG Media and Entertainment Industry Report [1] from 2017, by the end of 2016, there were around 6,000 single screens and around 2,500 multiplex screens in India. Multiplexes have been adding screens at the rate of 8-9% annually over the past few years, in 2016, multiplexes together added approximately 200 screens across the country and trends indicate that the industry is likely to continue to grow at a similar pace, adding 150-200 screens a year. Considering the enormous number of movie-goers and the amount of resources spent, there arises a need for optimizing the expenditure. Occupancy forecasting aims to minimize costs and maximize revenue, thus offering both public and private benefits.

Traditionally, revenue management is associated with statistical techniques which can predict, sometimes with good accuracy, occupancy rates and demand. However, some of the techniques require important statistical skills and lengthy procedures, particularly for Movie Occupancy Forecasting – which requires multiple Key Point Indicators (KPIs) and trend/behavior analysis to be applied in order for them to function accurately. Eliashberg et al. [2] talk about conditional forecasting and demand driven scheduling with traditional statistical approaches. Machine learning models for forecasting are relatively contemporary [3], and utilize learning algorithms to build predictive models for forecasting in the cinema industry – which is more fruitful, considering the amount of advances in the branches of big data and data science. These models can be easily used and deployed by personnel who do not possess much domain expertise or advanced training in statistics. Such models can be encapsulated into relatively inexpensive applications or computed on the cloud, thus providing cost-effective forecasts.

Forecasting is the most crucial component in pricing and budgeting [4], [5]. To develop optimal pricing strategies and minimize resources being spent, we explore and engineer features to capture user-behaviour and propose machine learning models for predictive forecasting, that predict using these learned features. Finally, feature tuning is applied to fine-tune the model according to the business model's requirements and use cases.

The key contributions of this paper are as follows:

- Proposing a historical transactional time-series dataset over five recent years for a top Indian movie multiplex chain.

2375-9259/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDMW.2019.00118

- Benchmarking new features exclusively for Movie Occupancy Forecasting by rigorous empirical experimentation and feature engineering.
- A study of various conventional machine learning and tree-based ensemble models, and state-of-the-art deep learning models including Recurrent Neural Networks (with Long Short Term Memory - LSTM).
- Proposing and optimizing on a novel *sale velocity* feature to capture the dynamic crowd behavior and demand over time.

The rest of the paper is organized as follows. We describe the transactional dataset and the ground truth utilized in Section II. Benchmarking and proposing various feature for Movie Occupancy Forecasting, the reasons behind using them and their importance, i.e, Feature Extraction and Engineering are discussed in Section III. The various machine learning, deep learning models used for time-series forecasting are presented in Section IV. The various metrics pertaining to movie occupancy forecasting are introduced and performance of the models are validated and elucidated in Section V. Section VI proposes a new dynamic feature to capture crowd behavior, and Section VII concludes the paper.

II. Dataset

The data is developed and housed by one of the renowned multiplex chains in India and is a collection of movie ticket booking transactions of a single multiplex sprawling over a period of five years from 2013 to 2017. It encompasses about a million transactions carried out in the following two ways - Offline (in-person ticket counters) and Online (with the corporation's website and mobile application). Each transaction comprises of a unique transaction ID, a movie ID, number of tickets booked, time of booking and other relevant metadata pertaining to the show and transaction. Historical transactional data of the company are stored data dumps (databases) from which the required data was extracted using SQL queries

TABLE I
Screen Capacity across 8 Screens

Screen	Capacity
S1	310
S2	310
S3	110
S4	110
S5	110
S6	131
S7	120
S8	120

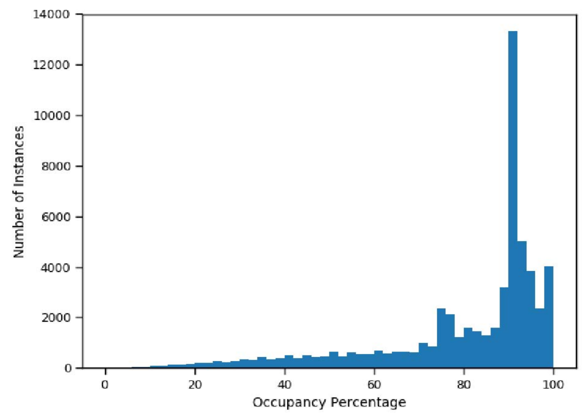
.

The multiplex spans across eight screens with different seating capacities as shown in Table I, operating with an average footfall of ∼5500 on holidays and ∼3500 on nonholidays, with each screen having at least four and/or five movies screened per day on holidays and nonholidays respectively.

Ground Truth (Percentage Tickets Sold - P_{sold})

P_{sold} represents the percentage of tickets sold or the final occupancy percentage of a show in the multiplex which is used as the ground truth for training and validating our forecasting model. Figure 1 shows the histogram of P_{sold} for all the screens from 2013 to 2017.

Fig. 1. Histogram of Occupancy percentages for shows from 2013 - 2017



Forecasting Requirements

A movie being played on a screen at a unique time is referred to as a show in this paper henceforth. The multiplex requires that the prediction of all shows for a day d_i should be a day-ahead i.e., predictions are made by time t_{pred} on the previous day d_{i-1}. Hence, for forecasting the occupancy of a given show, all the transactions occurring in [t_0, t_{pred}) are utilized, with t_0 denoting the time of the first booking made for the show.

III. Feature Engineering and Extraction

Feature engineering is the process of transforming raw data into features that enables the model to identify discriminant characteristics of the data, which results in improved model accuracies [6]. The performance of machine learning models is strongly dependent on the feature engineering phase. Therefore, the data pre-processing and transforming pipelines play a crucial role in deploying machine learning algorithms and is typically domain-specific involving considerable human expertise [7].

To represent a show, transactions leading up to the show cannot be directly fed into any machine learning model as they do not provide a unified representation of a show across characteristics like screen capacity, movie's prior performance, crowd behavior, etc. Hence, we develop features that best represent the shows of a day by leveraging empirical observations in the data, while taking the aforementioned characteristics of the domain into account.

The following relevant features for a movie occupancy forecasting paradigm are benchmarked and finalized by discussing with a domain expert, empirical analysis of the data as well

TABLE II
USEFUL NOTATIONS AND THEIR DESCRIPTION

Notation	Feature/Variable	Description
S_c	Screen Capacity	Total number of seats available i.e., total tickets available for sale
t_{sch}	Scheduled Screening Time	Time at which the show has been scheduled
t_0	Booking Start Time	Time at which the bookings for the show start
t_{pred}	Prediction Time	Time at which the occupancy has to be predicted
P_{held}	Percentage Tickets Held	Tickets unavailable for sale to the general public as percentage of S_c
P_{sold}	Percentage Tickets Sold	Tickets sold for a given show as a percentage of the S_c at t_{sch}
P_{avail}	Percentage Tickets Available	Tickets still available for sale at t_{pred}
P_{pred}	Predicted Ticket Sale Percentage	Tickets predicted to be sold as a percentage of S_c
n_t	Transaction Count	Total number of transactions made for the show until t_{pred}
T_{avg}	Average Tickets Sold per transaction	Mean tickets sold every transaction
v_x	Sale Velocity	Factor that accounts into screen capacity and time to sell $x\%$ tickets
s_x	Slot	Time Slot x of the day at which show is scheduled to be screened
t_{left}	Time Left To Show	Time left in hours for the screening of the show
d_r	Days Since Release	Days elapsed since the release of a show

as performance on the cross-validation data. The features and notations for the same can be observed in Table II.

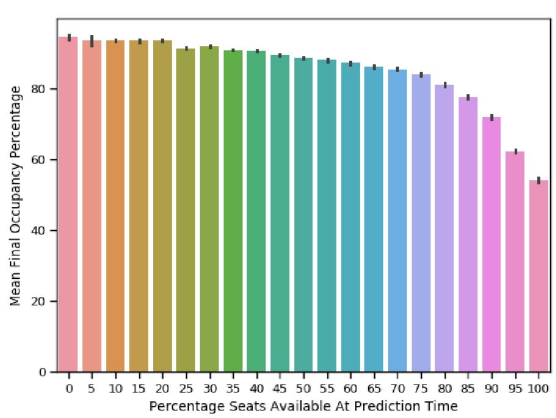
A. Percentage Tickets Held (P_{held})

While initiating ticket sales for a show, a small percentage of tickets denoted by P_{held} is reserved for the multiplex employees and/or its partners, and not open for sale to the general public. We observed that P_{held} directly influences the number of tickets sold, with a positive correlation.

B. Percentage Tickets Available (P_{avail})

The percentage seats available for sale at the time of prediction (t_{pred}) is denoted by P_{avail}. Figure 2 shows that availability of seats is inversely proportional to the total tickets sold (P_{sold}).

Fig. 2. Mean P_{sold} vs P_{avail}



C. Screen Capacity (S_c)

The capacity of a screen is denoted by S_c and their numeric values can be observed in Table I. We infer that for a show with higher S_c, it gets harder to achieve similar percentage of tickets sold (P_{sold}) than a show with lower S_c, since the remaining tickets to be sold are numerically higher.

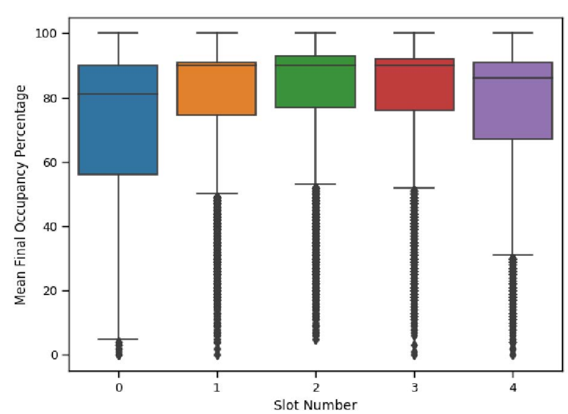
D. Slots (s)

The scheduled time of a show (t_{sch}) is observed to correlate with P_{sold}, since similar crowd-behavior across different days at particular times is observed. Hence, the general operating hours of the multiplex during which shows are played are discretized into five slots as shown in Table III, after experimenting with different number of slots. From the box plot in Figure 3, it can be inferred that shows being screened during s_0 (8 am - 11 AM) are erratic, with a high interquartile range (IQR) for P_{sold} which makes it difficult to model, while slots $s_{1\text{-}3}$ exhibited a more consistent turnout.

TABLE III
START AND END TIMES FOR DIFFERENT SLOTS

Slot	Start Time - End Time
s_0	08:00 - 11:00
s_1	11:00 - 14:00
s_2	14:00 - 17:00
s_3	17:00 - 20:00
s_4	20:00 - 01:00

Fig. 3. Box Plot - Relationship between s and Mean P_{sold}



E. Time Left to Show (t_{left})

Time remaining for the show to be screened at the time of prediction (t_{pred}), denoted by t_{left}, plays a vital role in tickets that will be sold in [t_{pred}, t_{sch}] (also referred to as *Walk-In Crowd* in the later sections of the paper). Figure 4 clearly shows that tickets sold after prediction (new tickets) are dependent on the time to show (t_{left}).

Fig. 4. ($P_{sold} - P_{AtPred}$) vs t_{left}

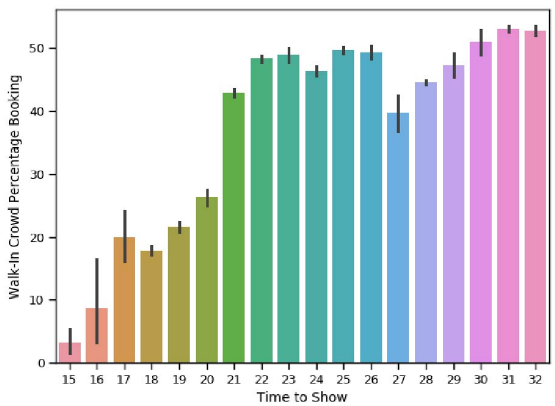
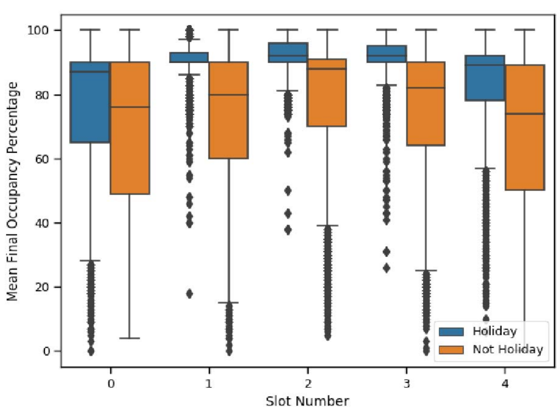


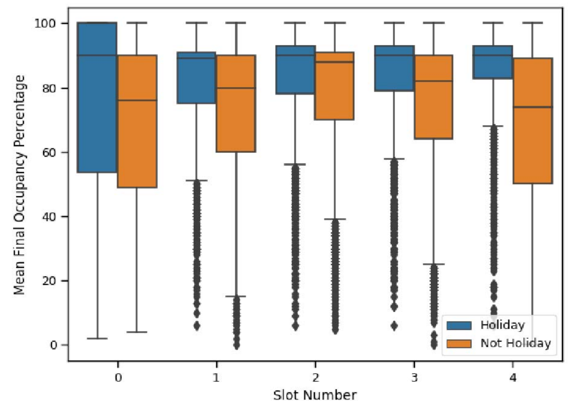
Fig. 5. Box Plot - Relationship between Show day holiday (h_s), Slot (s) and Mean P_{sold}



F. Holiday Factors (h_s and h_n)

The screening of a show in proximity of a holiday is vital in predicting P_{sold} and to model the same. Two binary features are added, namely - Show Day Holiday (h_s) and Next Day Holiday (h_n), where $h_s = 1$ when the show screening day is a holiday, while $h_n = 1$ when the day after the show's screening day is a holiday. These features are used as opposed to adding a 'day of the week' feature to account for holidays during the week. The relationship between h_s and h_n with Mean P_{sold} for shows in different slots (s) have been portrayed in Figures 5 and 6 respectively.

Fig. 6. Box Plot - Relationship between next day Holiday (h_n), Slot (s) and Mean P_{sold}

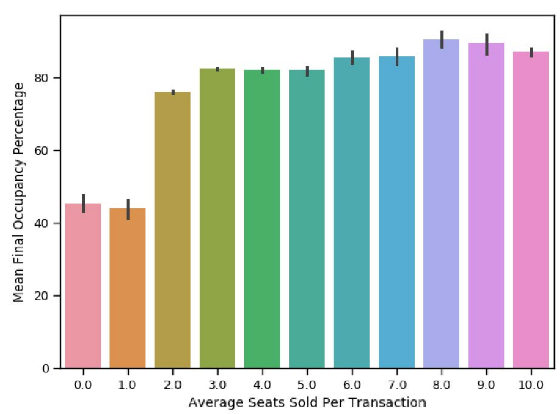


G. Average ticket sold per transaction (T_{avg})

Average number of seats or tickets sold per transaction of a show in [t_0, t_{pred}) is denoted by T_{avg} and is calculated as shown in Equation 1. It is a good indicator of the number of tickets that are likely to be sold in a single transaction that will happen in [t_{pred}, t_{sch}]. The T_{avg} is rounded off to the nearest integer. T_{avg} was observed to correlate with P_{sold} as shown in Figure 7.

$$T_{avg} = \left\lceil \frac{S_c - P_{AtPred}}{n_t} \right\rceil \tag{1}$$

Fig. 7. Box Plot - Correlation between T_{avg} and Mean P_{sold}



H. Days Since Release (d_r)

The feature *Days Since Release* (d_r) is the number of days since a particular movie was first shown and is capped at an empirically chosen maximum value of 200 days, with respect to Indian movie scenarios. We posit that d_r in tandem with the *History-related features* from Section III-I would capture a movie's occupancy pattern over time.

I. History-related Features

History-related features take into account the movie's performance prior to the predictions made for the given movie. Therefore, the most recent k shows were chosen for the same movie to extract historical features which are imperative in understanding the performance of a movie over time, which in turn effects the effectiveness of the forecasting model. These features include the last k historical percentage of tickets sold (P_{sold}), their mean (μ_h), median (m) and standard deviation (σ_h). The optimal value of k for this dataset is found to be 7 after rigorous experimentation on cross-validation data. If there are less than k screenings for the movie for which prediction is to be made, the missing values are set to -1 and are not included in the calculation of μ_h and σ_h.

IV. Models Architectures

We experiment across various ensemble tree-based, deep learning and ensemble models like variants of Gradient Boosting, Extremely Randomized Trees, variants of neural networks and LSTMs, and we describe only the following models which have performed the best. We utilize Grid Search for initial parametric optimization and arrive at the initial set of parameters, followed by introducing a new feature capturing crowd-behavior which is discussed in Section VI.

Conventional Movie Forecasting

Traditionally, the movie occupancy of the multiplex has been manually forecasted day-ahead by observing historical data using the domain expert's knowledge on the performance of the movies being played, their social media and public reception in tandem with reviews from film critiques. They make a prediction for an entire day as opposed to predicting for each show.

A. Extremely Randomized Trees (Extra Trees)

Conventionally, tree-based ensemble models have been effective in time-series paradigms. In this paper, we utilize an Extremely Randomized Trees (Extra Trees) Regressor model as proposed in [8], for the task of predicting P_{sold}. It also has an edge over Random Forests in terms of training and testing speeds. We utilize 100 estimators with minimum sample split and minimum leafs as 5, as the hyper parameters for the model.

B. Deep Neural Networks

Artificial Neural networks are a class of machine learning algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates [9]. A Deep Neural Network (DNN) contains layers of interconnected perceptrons, which is similar performing to multiple linear regressions. Neural nets are widely used because of their ability to generalize and respond to unexpected inputs/patterns. DNNs have the ability to learn and model non-linear and complex relationships and are also known as universal function approximators. The model consists of three dense layers with 128, 64, 32 neurons respectively, each with Leaky-ReLU as the activation function. Dropout with 0.2 probability between each Fully-Connected layer is introduced to reduce over-fitting [11], with the Huber loss function and RMSProp optimizer.

C. Branched LSTMs

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Networks (RNNs) capable of learning order dependencies in sequence prediction problems [10]. LSTMs have an edge over conventional feed-forward neural networks and RNNs in time-series forecasting problems, owing to their property of selectively remembering patterns over time, and their ability to learn required context to make predictions than being pre-defined.

In [10], it is also established with a variety of experiments that LSTMs can efficiently learn patterns sequential data and/or temporally sequential data. The predictions of n^{th} sample in a sequence of test samples can be influenced by an input present many time steps before. Gating mechanisms are responsible for preserving and releasing the long term dependencies in the network.

In our case, *Show History* is a time sequence feature which pertains to previous occupancies of the show. To include this feature effectively, the model takes in two separate inputs – show history and other features. We combine LSTMs for previous seven *Show History* data points, and Linear Dense layers for the other features resulting in a Branched-LSTM. The show history consists of the previous 7 history points and is directly fed into an LSTM branch. The LSTM branch consists of 3 layers with 128, 64 and 32 units respectively with a recurrent dropout of probability 0.3. The other features are fed into branch of 2 dense layers. The outputs from the LSTM branch and the Dense branch are concatenated with the other features and passed on to the Dense block. The Dense block consists of 3 layers with 128, 64, 32 units respectively. Activation function used is Leaky-ReLU with alpha value of 0.3, with the Huber loss function and RMSProp optimizer, similar to the DNN architecture observed in Section IV-B.

V. Metrics and Results

A. Metrics

In-order to validate the efficiency of the models mentioned in Section IV, we propose two metrics built on *Mean Absolute Percentage Error (MAPE)*, namely – ***Show-wise Error*** and ***Day-wise Error*** denoted by δ_S and δ_D respectively. δ_S for a show is computed as shown in Equation 2.

$$\delta_S = |P_{pred} - P_{sold}| \tag{2}$$

where, P_{pred} denotes the predicted sold ticket percentage.

We also compute Day-wise forecast to compare the model's performance against the corporation's current model mentioned in Section IV, with the model's day-wise error δ_D for a day D with shows S computed using Equation 3.

$$\delta_D = \frac{\left| \sum_{s \in S} P_{pred_s} S_{c_s} - \sum_{s \in S} P_{sold_s} S_{c_s} \right|}{\sum_{s \in S} P_{sold_s} S_{c_s}} * 100 \quad (3)$$

We also use *Mean Absolute Error (MAE)* defined as the average of the absolute differences in the actual turnout $P_{sold_s} S_{cs}$ and the predicted turnout $P_{pred_s} S_{cs}$ for all shows S of day D where $s \in S$. We utilize MAE for day wise predictions.

We also use E_{10}, the percentage of shows with MAPE < 10 for comparing the results along with $\mathbf{R^2}$score that is used to measure the performance of the regression models.

B. Data Split

The multiplex data is split into train, validation and test sets with shows from 2013-2016 used as the training set, first half of 2017 used for cross-validation and the second half of 2017 used for testing.

C. Show-wise and Day-wise Results

The show-wise & day-wise MAPE, MAE, and $\mathbf{R^2}$-score for the models discussed in Section IV have been shown in Tables IV and V respectively. It can be observed across all models that the Day-wise MAPE was quantifiably lower than the Show-wise MAPE which can be attributed to the fact that prediction errors from different shows of a day cancelled out each other. Furthermore, Extra-trees and B-LSTM models performed similarly with respect to the three metrics for both Show-wise and Day-wise predictions whereas, the same metrics calculated for DNN were sub-par in comparison.

Models	MAPE	$\mathbf{E_{10}}$	$\mathbf{R^2}$ Score
Extra Trees	7.70	73.74	0.92
DNN	9.02	67.91	0.89
B-LSTM	7.76	75.15	0.91

TABLE IV
SHOW WISE RESULTS

Models	MAPE	$\mathbf{E_{10}}$	MAE
Extra Trees	5.19	86.58	180.58
DNN	7.62	75.5	266.18
B-LSTM	5.54	84.46	200.37

TABLE V
DAY WISE RESULTS

VI. CAPTURING CROWD-BEHAVIOR

A. Sale Velocity (v_x)

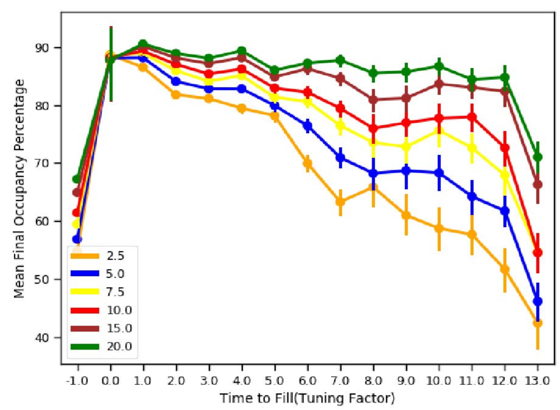
The model used only static features and does not take the demand for the show tickets into account, which can dynamically change over the course of the booking period thereby, leading to large δ_S as static features do not capture the changing demand.

We therefore use Sale velocity, denoted by v_x, to capture the rate at which the tickets are sold for a show. v_x indicates the ticket demand for the show and is calculated for different ticket sale percentages as a feature to capture the demand for the show. v_x can be defined as the ratio between $x\%$ of Screen Capacity (S_{cx}) and the difference between the time taken when $x\%$ of tickets are sold (t_x) and booking start time (t_0). The velocity of sale of $x\%$ tickets is calculated as shown in Equation 4.

$$v_x = \frac{S_{cx}}{t_x - t_0} \quad (4)$$

A screen with a seats may have higher velocity than a screen with b seats, given $a < b$, even when the same number of tickets have been sold. It can be inferred that S_c is directly proportional to the time taken to fill $x\%$ seats, hence the reason for utilizing S_c in the computation of v_x. If $x\%$ of tickets have not been sold at prediction time (t_{pred}), the sale velocity is set to -1. Figure 8 shows the correlation between mean percentage of final tickets sold and time to fill (t_{left}) for the aforementioned ticket-sale percentages.

Fig. 8. Line plot showing the correlation between Mean P_{sold} and t_{left} given v_x for different x (shown in the legend)



We can also observe that P_{sold} increases with increase in v_x for a given time. For instance, if a show has sold 20% of S_c in the same time as another show to sell 5% of S_c, then the former indeed must have a higher demand and thereby, higher occupancy. Hence, v_x is the dynamic feature which covers most ground when it comes to capturing crowd-behaviour.

Models	MAPE	$\mathbf{E_{10}}$	$\mathbf{R^2}$ Score
Extra Trees	7.71	73.9	0.92
ANN	8.52	71.65	0.90
B-LSTM	7.75	74.87	0.91

TABLE VI
SHOW WISE RESULTS WITH v_x

We incorporate Sale Velocity (v_x) as a feature with values of x as $[2.5, 5, 7.5, 10, 15, 20]$. From the results shown in Table VI & VII in comparison with Tables IV & V, it can be inferred that using v_x leads to a reduction in MAE for the day-wise predictions. However, the sale velocity is not available for all

Models	MAPE	$\mathbf{E_{10}}$	MAE
Extra Trees	5.14	87.58	179.23
DNN	5.04	86.58	187.63
B-LSTM	4.97	85.91	180.33

TABLE VII
DAY WISE RESULTS WITH v_x

the shows as $x\%$ of the tickets might not be sold for all values of x. We observed that the percentage of shows with error less than 10% increases with increase in x and this is can be used to indicate the error or prediction which might help to model the uncertainty.

VII. CONCLUSION

This paper discusses the problem of Movie Occupancy Forecasting in the present entertainment industry, and introduces new benchmarked features which can be effectively used to efficiently forecast the occupancy of a movie multiplex. Furthermore, the features engineered for this work are benchmarked and validated on tree-based ensemble machine learning models, and state-of-the-art deep learning architectures. The customer/crowd behavior over time was taken into account and a new *Sale Velocity* feature is proposed and optimized for maximizing the performance of the model. The aforementioned domain expert in Section IV performed with an MAPE of 8.04 which is outperformed by the models proposed. We believe our work can pave the way and act as a benchmark for effective Demand Forecasting, particularly – Movie Occupancy Forecasting systems.

VIII. ACKNOWLEDGMENT

The authors would like to acknowledge Solarillion Foundation for its support and funding of the research work carried out.

IX. FUTURE WORK

During real time prediction, the models discussed in the paper might become stale and subject to concept drift. One such mechanism to overcome this would be to supplement these models with incremental learning mechanisms that would ensure they stay up to date with the most recent and relevant trends. Also, movie occupancy predictions can be used in conjunction with other systems that address use cases such as dynamically pricing movie tickets and food sales forecasting [12].

REFERENCES

[1] *FICCI Media and Entertainment Report 2017*, KPMG India.

[2] Yue, X. and Liu, J., 2006. Demand forecast sharing in a dual-channel supply chain. European Journal of Operational Research, 174(1), pp.646–667.

[3] Caicedo, William and Payares, Fabin, *A Machine Learning Model for Occupancy Rates and Demand Forecasting in the Hospitality Industry*, 15th Ibero-American Conference on AI, San Jos, Costa Ric.,(2016), pp. 201–211.

[4] Lee, A.O., *Airline Reservations Forecasting: Probabilistic and Statistical Models of the Booking Process.*,Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA (1990).

[5] Weatherford, L.R., Kimes, S.E., *A Comparison of Forecasting Methods for Hotel Revenue Management*, International Journal of Forecasting 19(3), pp.401415 (2003).

[6] Jeff Heaton , *An Empirical Analysis of Feature Engineering for Predictive Modeling*, CoRR,abs/1701.07852, (2017).

[7] Yoshua Bengio and Aaron C. Courville and Pascal Vincent, *Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives*, CoRR, abs/1206.5538, (2012).

[8] Geurts, Pierre and Ernst, Damien and Wehenkel, Louis , *Extremely Randomized Trees*, Machine Learning, (2006), pp.3–42.

[9] Schmidhuber, J., *Deep Learning in Neural Networks: An overview*, Neural Networks,Volume 61, January 01, 2015, pp.85–117.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", Neural Computation, vol. 9, no. 8, pp.1735-1780, 1997.

[11] Srivastava, Nitish and Hinton, Geoffrey and Krizhevsky, Alex and Sutskever, Ilya and Salakhutdinov, Ruslan, *Dropout: A Simple Way to prevent Neural Networks from Overfitting*, The Journal of Machine Learning Research, (2014), pp.1929–1958.

[12] Adithya Ganesan V., Divi S., Moudhgalya N.B., Sriharsha U., Vijayaraghavan V. (2020) Forecasting Food Sales in a Multiplex Using Dynamic Artificial Neural Networks. In: Arai K., Kapoor S. (eds) Advances in Computer Vision. CVC 2019. Advances in Intelligent Systems and Computing, vol 944. Springer, Cham