Multi-task Envisioning Transformer-based Autoencoder for Corporate Credit Rating Migration Early Prediction

Han Yue  
hanyu@brandeis.edu  
Brandeis University  
Waltham, Massachusetts, USA

Steve Xia  
steve_xia@glic.com  
Guardian Life Insurance  
New York, New York, USA

Hongfu Liu  
hongfuliu@brandeis.edu  
Brandeis University  
Waltham, Massachusetts, USA

ABSTRACT
Corporate credit ratings issued by third-party rating agencies are quantified assessments of a company’s creditworthiness. Credit Ratings highly correlate to the likelihood of a company defaulting on its debt obligations. These ratings play critical roles in investment decision-making as one of the key risk factors. They are also central to the regulatory framework such as BASEL II in calculating necessary capital for financial institutions. Being able to predict rating changes will greatly benefit both investors and regulators alike. In this paper, we consider the corporate credit rating migration early prediction problem, which predicts the credit rating of an issuer will be upgraded, unchanged, or downgraded after 12 months based on its latest financial reporting information at the time. We investigate the effectiveness of different standard machine learning algorithms and conclude these models deliver inferior performance.

As part of our contribution, we propose a new Multi-task Envisioning Transformer-based Autoencoder (META) model to tackle this challenging problem. META consists of Positional Encoding, Transformer-based Autoencoder, and Multi-task Prediction to learn effective representations for both migration prediction and rating prediction. This enables META to better explore the historical data in the training stage for one-year later prediction. Experimental results show that META outperforms all baseline models.

CCS CONCEPTS
• Computing methodologies → Neural networks; • Applied computing → Enterprise data management.

KEYWORDS
Rating Migration, Fin-tech, Machine Learning

1 INTRODUCTION
Issuer credit ratings, developed by Moody’s in 1914 and by Poor’s Corporation in 1922, are one of the most important indicators of a corporation’s credit quality. After the issuance and assignment of the initial bond rating, these agencies regularly perform reviews of the underlying issues, which may result in a change in the rating being upgraded or downgraded. Altman [2], Härdle et al. [20], Saunders and Allen [49] point out that credit rating migration plays an integral part in the more general field of credit risk assessment of corporate bonds. Thus, corporation ratings or rating migration is one of the most crucial factors in investment decisions. Credit ratings are also a critical part of the modern financial regulatory framework because a regulated financial institution’s required capital (RBC, or risk-based capital) is directly linked to the quality of its assets, as measured by credit ratings.

Rating changes often have material impacts on the performance of an issuer’s debt. Being able to predict rating changes ahead of time has many applications, from better returns for an investor to the ability to better control and monitor risks for regulators. In this paper, we consider addressing the corporate credit rating migration early prediction challenge, which predicts whether the credit rating of a corporate issuer will be upgraded, remain unchanged, or downgraded after a period of time, such as 12 months.

In recent years, machine learning techniques are becoming popular in the financial area. For example, Ahelegbey et al. [1] and Chen and Tsai [10] use advanced machine learning and deep learning techniques to study financial risks. Guo and Li [19], Puh and Brkić [45], and Nur-E-Arefin and Mahmoud [41] explore the usage of machine learning for credit card fraud detection. Liébana-Cabanillas et al. [57], Liébana-Cabanillas et al. [36], and Hew et al. [25] build machine learning models in the banking operations domain. Villuendas-Rey et al. [56] and Xia et al. [58] work towards credit scoring. Lee [34] and Jabeur et al. [27] adopt machine learning methods for bond rating prediction. In this paper, we focus on the rating migration early prediction problem, that is, to forecast the migration of the bond rating of a corporate before it happens. To the best of our knowledge, there are no designated machine learning techniques for rating migration early prediction.

The rating migration early prediction problem is quite challenging. One of the main reasons is corporate credit ratings, issued by third-party rating agencies, are evaluated by taking many macro and issuer-specific factors into consideration. The process is impacted by both quantitative and qualitative considerations and sometimes complicated by the business relationships between the rating agencies and the issuers. Moreover, different rating agencies may have different opinions on the same issuer at any specific point.
in time. In this paper, we focus on solving the following two challenges: the first one is how to find the pattern hidden behind the highly dynamic corporate fundamental time-series data; the second challenge lies in early prediction. Since we aim to predict the rating migration of a period of time later, the corporate data during such a period are unavailable to be used for prediction, which further increases the difficulty.

To tackle the two challenges mentioned above, we propose Multi-task Envisioning Transformer-based Autoencoder (META) to address the rating migration early prediction problem. META consists of Positional Encoding, Transformer-based Autoencoder, and Multi-task Prediction to learn effective representations for both migration prediction and rating prediction. Specifically, we first adopt a Positional Encoding layer, which incorporates time information with corporate information to handle the time-series issue. Then we design a Transformer-based Autoencoder to learn envisioning ability from historical data, thus addressing the second challenge. Finally, we apply Multi-task Prediction to improve the migration task by performing a rating prediction task simultaneously. In summary, we highlight our contributions as follows:

- We consider a novel problem, corporate credit rating migration early prediction, which predicts the credit rating of a corporate will be upgraded, unchanged, or downgraded after a period of time. To our best knowledge, this crucial problem has not been explored in the literature.
- We propose a Multi-task Envisioning Transformer-based Autoencoder (META) model, which consists of Positional Encoding, Transformer-based Autoencoder, and Multi-task Prediction to learn effective representations for both migration prediction and rating prediction.
- Technically, we address the early prediction problem of the time gap between the training and prediction timestamp, where the corporate data during the gap period are unavailable in the training stage.
- Extensive experimental results demonstrate that our proposed META performs better than other baseline models. We also provide in-depth explorations of META with more interpretation in prediction tasks and the time gap between training data and test data.

2 RELATED WORK

In this section, we introduce the related work in terms of machine learning applications in the finance domain and techniques in time-series prediction and highlight the difference between our research problem and the early prediction in the literature.

Machine Learning in Financial Area. In the past decade, applications of machine learning techniques in finance have become steadily increasing [7, 24, 50]. Among diverse applications in finance, the prediction problem is one of the most common. Stock price prediction is one of the most studied financial applications. Vargas et al. [54] adopt a model composed of Convolutional Neural Network (CNN) and Long-Short Term Memory network (LSTM) to extract information from S&P500 index news for price prediction and intraday directional movement estimation. Das et al. [13] use a Recurrent Neural Network (RNN) together with sentiment analysis on Twitter for stock price forecasting. Zhou et al. [62] use a Generative Adversarial Network to minimize forecast error loss and direction prediction loss for stock price prediction. Some studies focus on predicting stock market indexes instead of prices. Dingli and Fournier [14] make use of CNN to forecast the next period direction of S&P500 index. Jeong and Kim [28] build a Reinforcement Learning model to predict S&P500 index. LSTM-based models with various other data [11, 39, 51] are developed for index prediction. Some researchers also focus on forecasting the underlying volatility of assets, where volatility is a statistical measure of the dispersion of returns for risk assessment and asset pricing. Doering et al. [15] implement a CNN model for volatility prediction. Zhou et al. [63] use LSTM and keywords from daily search volume based on Baidu to predict index volatility. Others [29, 40, 44] design generalized autoregressive conditional heteroscedasticity (GARCH)-type models for volatility prediction. There are also several studies on bond rating prediction. Lee [34] apply Support Vector Machine (SVM), and Kim [30] use ensemble learning to predict bond rating. Jabeur et al. [27] adopt the cost-sensitive decision tree algorithm [8] as well as SVM and Multi-Layer Perceptron (MLP) for bond rating prediction. While machine learning methods are widely applied in finance, the early prediction of rating migrations has not been explored as far as we know.

Time-Series Prediction. Time-series predicting plays an essential role in a wide range of real-life problems, thus leading to a huge body of works in this area. Many Recurrent Neural Network (RNN)-based methods [38, 47, 48, 57] have been developed for time-series prediction by modeling the temporal dependencies. LSTM [26] is one of the most popular RNNs, which addresses the problem of exploding and vanishing gradients by improving gradient flow with a cell, an input gate, an output gate, and a forget gate. Gated recurrent units [12] is similar to LSTM but without an output gate. DeepAR [48] combines autoregressive with RNNs and produces a probabilistic distribution of time series. Inspired by CNNs, which can capture local relationships, some researchers also design causal convolutions that use past information for forecasting. WaveNet [42] uses CNN to represent the conditional distribution of the acoustic features given the linguistic feature. TCN [4] combines residual connections with causal convolutions. LSTNet [33] uses CNN with recurrent-skip connections to capture temporal patterns. Recently, Transformer [55]-based architectures using attention mechanisms show great power in sequential data. Fan et al. [16] use attention to aggregate features extracted by Bi-LSTM. LogTrans [35] adopts causal convolution to capture local context and designed LogSparse attention to select time steps. Informer [61] extends Transformer with KL-divergence-based self-attention mechanism, distilling operation, and a generative style decoder.

Different from the methods mentioned above, we focus on a novel time-series prediction approach, which is predicting several steps ahead instead of predicting the following step immediately. Although early prediction problems in the data mining area have been addressed in several scenarios, including disease diagnosis [59, 60], student performance prediction [9, 46], action recognition [32, 53], and so on, these problems usually consider the time-series data that only access partial or preceding frames to recognize its static category along with all the timestamps. On the contrary, the problem we address here has dynamic labels, i.e., the same corporate credit ratings at different timestamps might be different.
3 DATA DESCRIPTION

In this paper, we use a dataset of historical commercial corporate ratings with a time length of 24 years ranging between 1997 and 2020. It includes a total of 445 unique companies. Data used contains two parts: corporate information and rating migrations, which we provide more details below. These data came from rating agencies.

Corporate information. Indexed by filing date and corporate identifier, the corporate information data contain 70 features, including 8 features in balance sheet, 2 features in capital structure, 3 features in cash flow, 8 features in income statement, 5 features in market data, 3 supplemental features, 6 features based on feature engineering, and 32 features based on ratios from long-term liquidity, short-term liquidity, profitability, margin analysis, and above-mentioned feature types. The interval of data for a same company is about 3 months in most cases.

Rating migrations. The ratings of companies in the dataset include 18 levels, ranging from high to low are: 'AAA,' 'AA+,' 'AA,' 'AA-,' 'A+,' 'A,' 'A-,' 'BBB+,' 'BBB,' 'BBB-,' 'BB+, ' 'BB,' 'BB-, ' 'B+,' 'B,' 'B-,' 'D,' 'NR.' There are two kinds of rating migration involved in the dataset: downgrade and upgrade. The downgrade means the rating of a company is changed to a lower level (higher risk), and the upgrade means the opposite. Different from corporate information data, the rating migration data is not seasonal due to the fact that rating agencies can change the rating at any time. Overall, there are 3384 rating migrations that happened during the time period, where 1964 of them are downgrades, and 1420 are upgrades.

Data pre-processing. To keep the dates and identifiers of both parts of the data consistent for usage, we match the ratings of corporates with the indices of the corporate information data and calculate a 12-month rating migration for each corporate based on the rating migration data in the next 12 months. We also go through several further steps for data pre-processing. Firstly, we normalize the data to eliminate the influence of feature magnitude. Secondly, we fill in zeros for the missing data to guarantee the functioning of algorithms. Thirdly, we remove data points with ratings of 'B,' 'B-,' and 'D' due to that the number of their migrations is less than 10. We also remove data points with ratings of 'NR' because the corresponding companies are not rated by rating agencies. Finally, we re-organize the corporate information into time-series data with an interval of 3 months (seasonal) based on corporate identifiers. For those who do not have enough previous data, we use the closest data point in time instead to make sure that all data points have the same dimension. After pre-processing, we get a total of 18674 data points with 14 levels of ratings, where the numbers of upgraded, downgrades, and unchanged are 1175, 2164, and 15335, respectively.

Visualization of statistics. Figure 1 shows statistics of the processed data. Figure 1(a) and Figure 1(b) plot the number and percentage of companies with 14 ratings from AAA to B+ by year from 1997 to 2020; (c) Percentage of upgrade and downgrade companies and the total number of companies in the same period; (d-e) Distributions of four corporate features in balance sheet and market among 4 rating groups in AAA, AA, A and BBB; (f) Rating migration heatmap among different rating groups.
We also demonstrate the distributions of total assets, total liabilities, market cap, and total enterprise value in balance sheet and market among 4 rating groups in AAA, AA, A, and BBB in Figure 1(d) & 1(e). The companies in AAA have much higher values in terms of the above four features than ones in other categories. The companies rated BBB have lower average but higher variances in these four features relative to higher-rated companies. Figure 1(f) shows the rating migration heatmap among 14 rating groups. The numbers in the dialog mean the unchanged ratios, which indicates the rating migration problem is an extremely imbalanced prediction problem. While considering the sizes of different rating groups, most rating migrations occur in A+, A, A-, BBB+, BBB, and BBB-.

4 PROBLEM FORMULATION

Rating migration is more relevant to long-term trading for investors compared to short-term trading. Given historical information on companies, the goal of the rating migration prediction problem is to predict how the rating of these companies will change after 12 months. The migration includes upgraded, downgraded, and unchanged. Thus it can be formulated as a multiclass classification problem with 3 categories. Let \( T \) denote the time range of historical data, \( D \) the dimension of corporate information, \( X \in \mathbb{R}^{T \times D} \) the time-series information of a corporate, and \( Y \in \{-1, 0, 1\}^T \) the corresponding rating migrations, and then the problem can be formulated as to find a mapping function \( f: X^N \rightarrow Y^N \), where \( N \) is the number of samples, such that a rating migration is predicted given historical data of a corporate.

In this prediction problem, there are two challenges we want to deal with. The first one is about the time-series nature of rating change decisions. The rating of a corporate is changed based on information available to the rating agencies with a strong historical perspective. This means that historical company performance, as well as the variances of its performance between different dates, have an impact on whether and how the rating will be changed. Therefore, it is necessary to capture the hidden momentum from historical time-series data in developing a model. The second challenge is lagged training data. Because our goal is to predict the rating migration of companies after 12 months, the corporate information data used for model training must be at least 1 year ago or even earlier to ensure the availability of labeling. It means that we are trying to predict the rating migrations of companies with the most recent data based on somewhat outdated data.

5 METHODOLOGY

In this section, we first introduce the framework of our proposed model, then elaborate on the objective function in detail.

5.1 Framework of META

We propose our model to address the challenges mentioned in Section 4. Figure 2 shows the framework of META, which mainly consists of three parts: Positional Encoding, Transformer-based Autoencoder, and Multi-task Prediction. META takes inputs from two periods, where data from the first period is what we want to predict migrations for, and data from the second period is the 1-year-lagged data of the first period. In order to handle the time-series historical data, we adopt Positional Encoding to generate representations with time information included for inputs of both periods. Then we build a Transformer-based Autoencoder to take representations of the first period as input, and we use Mean Squared Error to align the decoding output with representations of the second period, giving META the ability to envision 1-year-lagged data. While the 1-year-lagged data is only required in the model training stage and not needed in the test stage, all the historical data is used either as the first-period or as the second-period data in model training, and no extra data is required for the model test. Thus the issue of outdated data is addressed. Considering that the migration prediction task has a close relationship with the rating prediction task, the learning of the two tasks may help each other. Therefore, in the Multi-task Prediction part, we build a Common Embedding layer on top of the hidden layer of the Transformer-based Autoencoder to get common representations, then use a Migration Prediction layer and a Rating Prediction layer to get predictions for migrations and ratings, respectively, to achieve multi-task learning. Table 1 illustrates the notations along the following sections and their dimensions.

| Notations | Description | Dimension |
|-----------|-------------|-----------|
| \( T \)   | Time range of historical data | Scalar    |
| \( N \)   | Number of samples | Scalar    |
| \( D \)   | Dimension of company information | Scalar    |
| \( X \)   | Historical information of a company | \( T \times D \) |
| \( X \)   | 1-year-lagged information of \( X \) | \( T \times D \) |
| \( H \)   | Positional encoded embedding for \( X \) | \( T \times D \) |
| \( H \)   | Positional encoded embedding for \( X \) | \( T \times D \) |
| \( Z \)   | Hidden features of Transformer-based Autoencoder | \( T \times 256 \) |
| \( A \)   | Output of Transformer-based Decoder | \( T \times D \) |
| \( M \)   | Prediction for migration after 1 year | \( T \times 3 \) |
| \( R \)   | Prediction for rating after 1 year | \( T \times 14 \) |
| \( Y_M \) | One-hot ground truth of migration after 12 months | \( T \times 3 \) |
| \( Y_R \) | One-hot ground truth of rating after 12 months | \( T \times 14 \) |

5.2 Objective Function

Our model consists of three parts: Positional Encoding, Transformer-based Autoencoder, and Multi-task Prediction. In this subsection, \( X \) and \( \hat{X} \) represent input data of the first and second (1-year-lagged) period, respectively. We use \( \theta = \{\theta_E, \theta_D, \theta_P\} \) to denote the trainable parameter set of Transformer-based Encoder, Transformer-based Decoder, and Multi-task Prediction in the proposed model. Specifically, \( \theta_E = \{\theta_O, \theta_Q, \theta_K, \theta_V\} \) denotes the trainable parameter set in the Transformer-based Encoder part, where \( \theta_O, \theta_Q, \theta_K, \) and \( \theta_V \) are the trainable parameters of the linear layer, queries, keys, and values in the multi-head attention layer, respectively, and \( \theta_F \) is the trainable parameters of the fully connected layer. \( \theta_P \) denotes the trainable parameter set in the Transformer-based Decoder part, which has similar components as \( \theta_E \) but different dimensions. \( \theta_P = \{\theta_C, \theta_M, \theta_R\} \) denotes the trainable parameters in the Prediction part, where \( \theta_C, \theta_M, \) and \( \theta_R \) are the trainable parameters of the Common Embedding, Migration Prediction, and Rating Prediction, respectively. Our goal is to minimize the objective function by adjusting \( \theta \) with the model and data. Each part of the model is detailed as follows.

**Positional Encoding.** Positional Encoding [55] is designed for the model to make use of the order of the sequence without involving recurrence and convolution. To inject some information about the relative or absolute position of the time-series data, we adopt a fixed positional encoding method [18], which is sine and cosine
Autoencoder is designed to translate the current data into 1-year-lagged data, where the hidden features with this envisioning are generated. The Transformer-based Encoder has two sub-layers. The first sub-layer is the position, and the second sub-layer is the position-wise, fully connected feed-forward network. There is a multi-head self-attention mechanism, and the second is a self-attention mechanism, which allows the model to jointly attend to information from different representation subspaces at different positions, which can be formulated as:

$$E = \text{Concat}(h_1, h_2, ..., h_h) \theta_O, \quad \text{head}_i = \text{Attention}(H \cdot \theta_Q, H \cdot \theta_K, H \cdot \theta_V), i \in [1, h],$$

where $H$ denotes the number of heads, and $\theta_Q, \theta_K = \{\theta_{Q_1}, ..., \theta_{Q_h}\}$, $\theta_V = \{\theta_{V_1}, ..., \theta_{V_h}\}$ are learnable parameters. With a layer normalization function $\text{LayerNorm}(\cdot)$ [3], the output of the first sub-layer in Transformer-based Encoder can be written by:

$$E = \text{LayerNorm}(H + E).$$

Then for the second fully connected layer together with another normalization layer, the Transformer-based Encoder generates a hidden representation $Z$ by the following equation:

$$Z = \text{LayerNorm}(\tilde{E} + \tilde{E} \cdot \theta_F),$$

where $\theta_F$ denotes the learnable parameters in the fully connected layer.
Similar to the encoder part, the Transformer-based Decoder also contains a multi-head self-attention layer as well as a position-wise fully connected layer. To make it easy, we use $Z = Encoder(H, \theta_{E})$ to denote the operations and outputs of the encoder part, and $A = Decoder(Z, \theta_{D})$ the decoder part.

**Multi-task Prediction.** The migration prediction task is highly related to the rating prediction task because the migrations can be inferred by ratings. To help improve the performance of migration prediction, we design a multi-task learning method, which can forecast migrations and ratings simultaneously. To achieve this, we first build a common embedding layer to generate representations for both prediction tasks, which can be written as:

$$C = \text{ReLU}(Z \cdot \theta_{C}),$$

where $\text{ReLU}(x) = \max(x, 0)$ is the activation function, and $\theta_{C}$ denotes the learnable parameters. Then we build the migration prediction part and rating prediction part on top of the common embedding layer as follows:

$$M = \text{Softmax}(C \cdot \theta_{M}),$$

$$R = \text{Softmax}(C \cdot \theta_{R}),$$

where $M$ is a 3-dimensional vector denoting the probabilities of upgraded, unchanged, and downgraded, and $R$ is a 14-dimensional vector denoting the probabilities of under different rating levels after 12 months.

**Overall Objective Function.** Our objective function contains 3 parts. The first one is the mean squared error of $A$ and $\hat{H}$, driving the model to learn an envisioning ability in the Transformer-based Autoencoder part, which is:

$$L_{A} = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{D} (A_{t,i} - \hat{H}_{t,i})^2,$$

where $T$ is the time length of historical data, and $D$ is the dimension of input features. The other two parts of the objective function are cross-entropy losses for migration prediction and rating prediction:

$$L_{M} = \frac{1}{T} \sum_{t=1}^{T} \sum_{p=1}^{3} (1 - Y_{Mt,p}) M_{t,p},$$

$$L_{R} = \frac{1}{T} \sum_{t=1}^{T} \sum_{p=1}^{14} (1 - Y_{Rt,p}) R_{t,p},$$

where $Y_{M}$ and $Y_{R}$ are one-hot encodings of ground truth for migration and rating, respectively.

Combining Eq. (12), Eq. (13), and Eq. (14), our overall objective function is:

$$\min_{\theta} L_{A} + \alpha L_{M} + \beta L_{R},$$

where $\alpha$ and $\beta$ are hyperparameters controlling the weights of $L_{M}$ and $L_{R}$. We adopt Adam optimizer [31] to minimize the objective function. More detailed settings can be found in the following section.

### 6 EXPERIMENTS

In this section, we first introduce the experimental setting, then we evaluate our model. Finally, we provide some insightful experiments to demonstrate the effectiveness of the proposed model.

#### 6.1 Experimental Setting

In our experiments, we aim to predict rating migrations after 12 months for companies. We also set the historical data length of each data point to be exactly 12 months to meet the prediction time range. The total test period is from 2005/01/01 to 2020/12/31. We adopt an expanding window for the training set and re-calibrate the models every 3 months. For example, we train the model with data from 1997/01/01 to 2004/12/31 (available migration labels are from 1997/01/01 to 2003/12/31) and use the trained model to predict the 12-month-later migration of a corporation with its information data on and before 2005/01/01. Here the ground truth of 12-month-later migration is decided on 2006/01/01.

**Baseline methods.** We choose 6 classical classification models as baseline methods for comparison: K-NearestNeighbor (KNN) [17], Support Vector Machine (SVM) [5], Random Forest (RF) [6], Multi-Layer Perceptron (MLP) [22], AdaBoost [21], and Gaussian Naïve Bayes (NB) [52]. These methods are implemented based on Sklearn. Among them, SVM, MLP, and ensemble methods like AdaBoost are adopted by previous studies [27, 30, 34] on rating prediction. Beyond these, we also choose 5 time-series prediction models for comparison: Long Short-Term Memory (LSTM) [26], DeepAR [48], Transformer [55], LogTrans [35], and Informer [61]. LSTM is implemented based on Keras, Transformer is implemented based on Pytorch. For the rest of methods, we use the public source codes provided by the authors on Github.

**Implementation details.** For the proposed Multi-task Envisioning Transformer-based Autoencoder (META), we implement it by Pytorch [43], adopt Adam optimizer [31] with a learning rate of 0.0001, read the data with a batch size of 1024, and run 3000 epochs for each re-calibration to train the model. The values of $\alpha$ and $\beta$ in the objective function are set to 1 by default.

**Evaluation metric.** While the numbers of upgraded, downgraded, and unchanged companies are imbalanced as shown in Figure 1(c), we decide to use $F_1$ score for both upgrades and downgrades as our evaluation metric. In supplement to $F_1$ score, we also provide accuracy to verify that the models are not performing a poor prediction for unchanged situations. The calculations of $F_1$ score and accuracy are as follows:

$$\text{precision} = \frac{tp}{tp + fp}, \quad \text{recall} = \frac{tp}{tp + fn}$$

$$F_1 = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}},$$

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + fn + tn},$$

where $tp$, $fp$, $fn$, $tn$ are the numbers of true positives, false positives, false negatives, true negatives, respectively. Here we take $F_1$ score for upgrades as an example, then $tp$ denotes the number of upgraded samples that are predicted correctly by the model, $fp$ denotes the number of downgraded or unchanged samples that are predicted incorrectly by the model, $fn$ denotes the number of upgraded samples that are predicted incorrectly by the model, and
is the number of downgraded or unchanged samples that are predicted correctly by the model.

Since the early rating migration problem is an extremely imbalanced prediction problem, F1 is more suitable than Accuracy for algorithmic evaluation. Moreover, in this financial application, we care more about the downgrade migration, i.e., F1-Down, because of its strong implication for not only investment professionals but also risk managers and regulators.

### 6.2 Algorithmic Performance

Based on the settings mentioned above, we run all the methods and gather their predictions from 2005/01/01 to 2020/12/31. Table 2 shows the overall performance of all models during the test period. Among classical non-time-series baseline models, SVM achieves the highest accuracy but gets very low F1-Up and F1-Down. Furthermore, SVM gives a very high number of unchanged predictions, which happens to achieve the best accuracy on this unbalanced dataset where 85.77% samples are unchanged. This is also a good example to illustrate why we choose F1-Up and F1-Down for evaluation. Another interesting result is achieved by NB, whose accuracy is low at 21.60%, indicating that NB is trying not to predict unchanged situations for samples, and the F1-Up and F1-Down are somehow good because it achieves a high recall. The time-series algorithms, including two classical deep models and three state-of-the-art models, deliver similar performance as non-time-series methods. The reason is that there is a 1-year-gap between their training and test data, and the data distribution may change during this period. On the contrary, META leverages the influence of this gap by adopting positional encoding and autoencoder to use the data during the gap period. Therefore, the representation learned by META is capable of containing future trends for early prediction. It not only handles the time-series data but also has the envisioning ability, which makes full use of all data in the training set no matter whether the data is labeled or not. Moreover, different from baseline methods, META is the only model that performs multi-task learning, which also helps improve the model in providing stable and reliable predictions. Compared with baseline methods, our proposed META outperforms and has a significant advantage over the baseline and state-of-the-art methods on both F1-Up and F1-Down.

Figure 3 shows the detailed early prediction performance of three state-of-the-art time-series algorithms and META by year and by rating group, where the years in recessions and early-recessions are highlighted with light blue and dark blue shadow, and five rating groups with frequent migrations are chosen. META outperforms other competitive methods by large margins in most years, which demonstrates the effectiveness of our proposed method for early prediction. Another noteworthy observation is that META performs even better in predicting downgrade risk in the time periods right before and around recessions. In financial applications, a higher early predictive capability is particularly beneficial right before major economic and market downturns. To this end, META performs much better in the recession and early-recession periods than in normal periods, which can be regarded as a significant advantage in practical use. Moreover, if we take a close look at the rating groups (A+, A, A-, BBB+, BBB, and BBB-), where rating migrations frequently occur and are of great value in financial investigation, our META also outperforms other methods as well.

### 6.3 In-depth Exploration

Beyond the above algorithmic comparisons with several baseline methods, we also provide in-depth explorations of our proposed META in terms of multi-task prediction, envisioning verification, and different gap periods for early prediction.

**Multi-task Prediction.** The proposed META is a multi-task model, and it generates two kinds of predictions: migration and rating. While inferring from the rating prediction result, we can easily get another migration result that may not be the same as the migration prediction by META directly. Table 3 shows our exploration of META in performing different tasks and applying different methods.
When performing both tasks together, the migration prediction demonstrates the effectiveness of META.

### Table 3: Performance of META under different tasks. Checkmark denotes that the corresponding task is performed. Results from rating to migration mean that the migration is calculated based on the rating prediction results of META, and the others are directly predicted by META.

| Migration | Rating | Results From | \(F_1\) Up | \(F_1\) Down | Accuracy |
|-----------|--------|--------------|--------------|--------------|----------|
| ✓ ✓ | Rating → Migration | 0.1944 | 0.1211 | 0.3874 |
| ✓ ✓ | Directly Predicting Migration | 0.9577 | 0.1203 | 0.7724 |
| ✓ ✓ | Rating → Migration | 0.1278 | 0.1561 | 0.6088 |
| ✓ ✓ | Multi-task Predicting Migration | 0.1864 | 0.2738 | 0.7909 |

Table 4: Algorithmic performance of 6 non-time-series and 6 time-series methods on no gap rating migration prediction from 2005 to 2020 by \(F_1\)-Up, \(F_1\)-Down and Accuracy.

| Method      | \(F_1\) Up | \(F_1\) Down | Accuracy |
|-------------|------------|--------------|----------|
| Adaboost    | 0.0525     | 0.1112       | 0.8164   |
| KNN         | 0.1752     | 0.2285       | 0.7693   |
| MLP         | 0.1429     | 0.2317       | 0.7554   |
| NB          | 0.1535     | 0.2014       | 0.7555   |
| RF          | 0.1644     | 0.2533       | 0.7116   |
| SVM         | 0.0000     | 0.0068       | 0.8332   |
| LSTM        | 0.0545     | 0.0902       | 0.8059   |
| Transformer | 0.1410     | 0.1748       | 0.7349   |
| DeepAR      | 0.1221     | 0.2210       | 0.7486   |
| LogTrans    | 0.1302     | 0.2083       | 0.7556   |
| Informer    | 0.1716     | 0.2415       | 0.7918   |
| META (no gap) | 0.2174   | 0.2812       | 0.7966   |
| META (12-month gap) | 0.1864   | 0.2738       | 0.7909   |

Figure 4: Early migration prediction of four time-series algorithms with different gap periods.

### 7 CONCLUSION

In this paper, we focus on the rating migration prediction problem. With historical information on companies, we aim to predict how the rating of these companies will change (upgraded, downgraded, and unchanged) 12 months later. We propose Multi-task Envisioning Transformer-based Autoencoder (META) to address this problem. Specifically, we first adopt a Positional Encoding layer to incorporate time information with company information, then we design a Transformer-based Autoencoder to learn envisioning ability from historical data. Finally, we use Multi-task Prediction to get predictions for both migrations and ratings simultaneously. Experimental results show that META outperforms all 11 baseline methods, which demonstrates the effectiveness of our proposed method. Our proposed model also has the advantage of higher precision during time periods that are most relevant for its applications, i.e., before and during economic recessions.

### REFERENCES

[1] Daniel Felix Ahelegbey, Paolo Giudici, and Branka Hadij-Misheva. 2019. Latent factor models for credit scoring in P2P systems. *Physica A: Statistical Mechanics and its Applications* 522 (2019), 112–121.

[2] Edward I Altman. 1998. The importance and subtlety of credit rating migration. *Journal of Banking & Finance* 22, 10-11 (1998), 1231–1247.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. arXiv preprint arXiv:1607.06450 (2016).

[4] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018).

[5] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Annual Workshop on Computational Learning Theory*.

[6] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[7] Longheng Cao, Qiang Yang, and Philip S Yu. 2021. Data science and AI in FinTech: An overview. *International Journal of Data Science and Analytics* 12, 2 (2021), 83–99.

[8] JH Chauchat, Ricco Rakotomalala, M Carlouz, and C Pelletier. 2001. Targeting customer groups using gain and cost matrix: a marketing application. *International Conference on Service Systems and Service Management*.

[9] Fu Chen and Ying Cui. 2020. Utilizing Student Time Series Behaviour in Learning Analytics. *Journal of Management Systems for Early Prediction of Course Performance*.

[10] Jia-Hao Chen and Yun-Cheng Tsai. 2020. Encoding candlesticks as images for pattern classification using convolutional neural networks. *Financial Innovation* 6 (2020), 1–19.

[11] Yuzhou Chen, Junji Wu, and Hui Bu. 2018. Stock market embedding and prediction: A deep learning method. In *International Conference on Service Systems and Service Management*.

[12] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014).

[13] Sushree Das, Ranjan Kumar Behera, Santanu Kumar Rath, et al. 2018. Real-time sentiment analysis of twitter streaming data for stock prediction. *Procedia Computer Science* 132 (2018), 956–964.
