Beyond Gut Feel: Using Time Series Transformers to Find Investment Gems

Lele Cao[1(✉)], Gustaf Halvardsson[1,2], Andrew McCornack[1], Vilhelm von Ehrenheim[1,3], and Pawel Herman[2]

[1] Motherbrain, EQT Group, Regeringsgatan 25, 11153 Stockholm, Sweden
caolele@gmail.com,
{gustaf.halvardsson,andrew.mccornack,vilhelm.vonehrenheim}@eqtpartners.com
[2] KTH Royal Institute of Technology, Lindstedtsvagen 5, 11428 Stockholm, Sweden
paherman@kth.se
[3] QA.tech, Gävlegatan 16, 11330 Stockholm, Sweden

Abstract. This paper addresses the growing application of data-driven approaches within the Private Equity (PE) industry, particularly in sourcing investment targets (i.e., companies) for Venture Capital (VC) and Growth Capital (GC). We present a comprehensive review of the relevant approaches and propose a novel approach leveraging a Transformer-based Multivariate Time Series Classifier (TMTSC) for predicting the success likelihood of any candidate company. The objective of our research is to optimize sourcing performance for VC and GC investments by formally defining the sourcing problem as a multivariate time series classification task. We consecutively introduce the key components of our implementation which collectively contribute to the successful application of TMTSC in VC/GC sourcing: input features, model architecture, optimization target, and investor-centric data processing. Our extensive experiments on two real-world investment tasks, benchmarked towards three popular baselines, demonstrate the effectiveness of our approach in improving decision making within the VC and GC industry.

Keywords: Company success prediction · Venture capital · Growth equity · Private equity · Investment · Multivariate time series

1 Introduction

Private Equity (PE) is a rapidly growing segment of the investment industry that manages funds on behalf of institutional and accredited investors. PE firms acquire and manage companies with the goal of achieving high, risk-adjusted

Gustaf Halvardsson (currently works for BCG GAMMA, part of BCG X) contributed (equally as Lele Cao) to this work as part of his Master's Thesis project carried out in EQT Motherbrain supervised by Lele Cao (industrial supervisor) and Pawel Herman (academic supervisor).

M. Wand et al. (Eds.): ICANN 2024, LNCS 15024, pp. 373–388, 2024.
https://doi.org/10.1007/978-3-031-72356-8_25

returns through subsequent sales [8]. These acquisitions can involve majority shares of private or public companies, or investments in buyouts as part of a consortium. Common PE investment strategies, as identified by [6], include Venture Capital (VC), Growth Capital (GC), and Leveraged Buyouts (LBO). These strategies offer varying degrees of risk and return potential, depending on the investment objectives and time horizon of the PE fund. The ability to accurately assess the likelihood of company success is crucial for PE firms in identifying attractive investment targets. Traditional evaluation of company performance often relies on manual analysis of financial statements or proprietary information, which may not be sufficient for capturing the dynamic nature of companies, especially those in early-stage or high-growth industries. This evaluation approach is often time consuming and as a result not every potential company can be properly evaluated. Therefore, there is a growing interest in leveraging data-driven methods to (1) debias decisions, so that the individual investment decision made for a particular deal is expected to drive lower risk and higher ROI (return on investment); and (2) enable automation, so that more companies can be evaluated without the need for additional resources [9].

For LBO in the PE industry, data-driven approaches may be less relevant due to the combination of two reasons: (1) LBO professionals often track and maintain in-depth knowledge of late-stage companies[1] in a few focus sectors, resulting in unique knowledge and understanding that can hardly be entirely replaced by public (or even proprietary) data; (2) the number of LBO investments is usually less than VC and GC leading to a lower sourcing frequency. VC investments often involve early-stage companies with prone-to-change business models and limited revenue, making data-driven approaches valuable for evaluating their growth potential. Additionally, VC investors typically manage larger portfolios with higher investment frequency, necessitating the use of data-driven models for efficient decision-making in identifying and evaluating investment opportunities. In practice, historical financial data (e.g., revenue) of startup[2] or scaleup[3] companies are commonly perceived as a good approximation of their true valuations [10]. The financial information of GC targets (scaleups) is much more accessible than that of VC targets (startups). Therefore, GC practitioners often directly use financial metrics to calculate the company's valuation for sourcing, which is why the adoption of big data in GC sourcing may not be as intensive as in VCs. However, data-driven approaches may still provide additional insights in assessing the growth potential and financial performance of the GC targets.

[1] Generally, a company is considered late-stage when it has proven that its concept and business model work, and it is out-earning its competitors.

[2] A startup is a dynamic, flexible, high risk, and recently established company that typically represents a reproducible and scalable business model. It provides innovative products or services, and has limited funds and resources [5,9,35,37].

[3] A startup moves into scaleup territory after proving the scalability and viability of its business model and experiencing an accelerated cycle of revenue growth. This transition is usually accompanied by the fundraising of outside capital [11].

Our contributions significantly advance data-driven strategies for sourcing investment opportunities in the VC and GC sectors by predicting the potential success of companies. These advancements include:

- We formally define the sourcing problem for VC/GC investments as a multivariate time series classification task and propose to employ a Transformer-based Multivariate Time Series Classifier (TMTSC) to address it.
- We introduce key components of our implementation, including input features, model architecture and optimization target, which all contribute to the successful application of TMTSC in VC/GC sourcing.
- We carry out extensive experiments, comparing TMTSC with widely adopted baselines on two real-world tasks, and demonstrate the effectiveness of our approach using a diverse set of evaluation metrics and strategies.

2 Related Work

Over the past two decades, data-driven approaches have been dominating research on deal sourcing for VC, i.e. identifying startups that eventually turn into unicorns[4]. In recent years, however, research has begun to intensify on GC deal sourcing, transforming the way scaleup companies are identified and assessed. Based on our extensive literature survey, data-driven methods for VC/GC deal sourcing can be broadly categorized into Statistical and Analytical (S&A) methods, conventional Machine Learning (ML) methods, and Deep Learning (DL) methods. S&A work [22,27,32,35] typically starts with defining some hypotheses followed by testing them using statistical tools. However, developing effective hypotheses for S&A approaches is a challenging task that requires simplicity, conciseness, precision, testability, and most importantly, a grounding in existing literature or established theory, as emphasized in [41]. It is worth mentioning that while DL methods technically fall under the broader umbrella of ML, we discuss DL work separately in recognition of its increasing popularity and relevance to our research.

2.1 Conventional Machine Learning Methods

Over the last few years, there has been a growing interest in leveraging ML algorithms for *hypothesis mining* from data, as an alternative to manually defining hypotheses upfront. Hypothesis mining involves conducting explainability analysis on trained ML models to summarize, rather than explicitly define, hypotheses [24]. For instance, by training an ML model on a labeled dataset containing features of various companies, and quantifying how changes in these features impact the prediction target (i.e. success probability), one can distill hypotheses that describe the relationships between the relevant features and the prediction

[4] Unicorn and near-unicorn startups are private, venture-backed firms with a valuation of at least $500 million at some point [14].

target. Compared to S&A, hypothesis mining is a much more structured procedure that trains an ML model using the entire dataset at hand. In general, ML based approaches, as demonstrated in previous works such as [3,7,27,29,43], typically require practitioners to define the input data $\mathbf{x}$ and annotation y (labeling "good" or "bad" investment according to some criteria) before training a model $f(\cdot)$ that maps $\mathbf{x}$ to y, i.e., $y = f(\mathbf{x})$. With the rapid growth of dataset size and diversity (origin and modality), conventional ML models[5] sometimes struggle to fit the large and unstructured[6] data due to lack of *capacity* and *expressivity*[7].

2.2 Deep Learning Methods

Most recently, DL algorithms have attracted an increasing number of researchers hunting for good VC/GC investment targets. DL is implemented (entirely or partly) with ANNs (artificial neural networks) that utilize at least two hidden layers of neurons. The *capacity* of DL can be controlled by the number of neurons (width) and layers (depth) [23]. Deep ANNs are exponentially *expressive* with respect to their depth [34]. While structured data is commonly used in many DL methods, such as [2,4,18], unstructured data is increasingly recognized as an important complement to structured data in recent studies [12,20,26,31,38], or even as a standalone input to the model [39,45]. Unstructured data often contains large-scale and intact-yet-noisy signals, which may result in superior performance when a proper DL approach is applied [19].

The main types of unstructured data seen include text [12], graph [1], image [13], video [39], audio [36] and time series [12]. Among these, fine-grained multivariate time series, which encompass various aspects of a company over time, hold particular significance for deal sourcing in the VC/GC domain. Some examples of these aspects include financial performance, team dynamics, funding rounds, market conditions, and other key indicators. Especially for GC, financial time series become highly relevant for evaluating scaleup companies whose periodical financial data points are usually available to the potential investor [10]. Due to the proprietary, costly, and scarce nature of multivariate time series company data, there is a limited number of DL based approaches in the literature that utilize time series as model input. To the best of our effort, we identified only three such studies [12,26,38], highlighting the challenges associated with utilizing multivariate time series data to source investment targets for Venture and Growth Capital. Inspired by [44], we frame the problem as a multivariate time series classification task and propose a solution that leverages a Transformer model. Our approach also incorporates carefully designed input features, optimization target, and investor-centric data processing [9].

[5] The frequently applied conventional ML models include many such as decision tree [3], random forest [29], logistic regression [27], and gradient boosting [43].

[6] Unstructured data, such as image and timeseries, is a collection of many varied types that maintains their native form, while structured data is aggregated from original (raw) data and is usually stored in a tabular form.

[7] *Expressivity* describes the classes of functions a model can approximate, and *capacity* measures how much "brute force" the model has to fit the data.

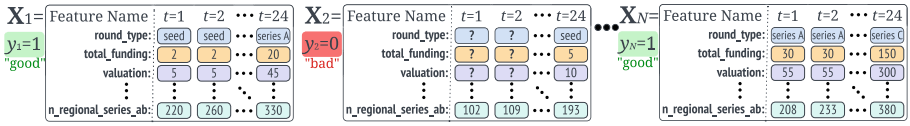


Fig. 1. An illustration of multivariate time series dataset (N samples) to train models for VC and GC sourcing.

3 The Approach

Our approach tackles the problem of identifying good investment targets for VC and GC by framing it as a multivariate time series classification task. Specifically, each potential investment target (i.e. candidate company) is represented by a multivariate time series $\mathbf{X} \in \mathbb{R}^{T \times K}$, as shown in Fig. 1. $\mathbf{X}$ consists of T observations, each containing K variables that describe different aspects (e.g., funding, revenue, etc.) of the corresponding company. Formally, each sample $\mathbf{X}$ is a sequence of T feature vectors: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T]$, where $\mathbf{x}_t \in \mathbb{R}^K$. At each time step t, we collect K numerical or categorical features about the company to form the vector $\mathbf{x}_t$, which captures a multi-view snapshot of the company at that time point. The last vector $\mathbf{x}_T$ represents the most recent state of the company. Depending on the data available, the time interval between two adjacent time points, t and $t+1$, can be set to a month, a quarter, or any other length of choice. By adopting this representation, we can model a multi-view evolution of each company over time and make informed predictions about their future success.

We collect a set of N samples, each corresponding to a company, denoted by $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, \ldots, \mathbf{X}_N$. For each sample $\mathbf{X}_n$, we have a binary ground truth label $y_n \in \{0, 1\}$ indicating a "bad" or "good" investment target according to some criteria. Details of how we define and collect these labels are explained in Sect. 3.3. We construct a dataset $\mathfrak{U}$ from these samples and labels as $\mathfrak{U} = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_N, y_N)\}$, where $n \in \mathbb{Z} \cap [1, N]$. Our objective then is to **train a model on $\mathfrak{U}$ to accurately predict the ground truth labels y_n using $\mathbf{X}_n$**. We use $\hat{y}_n \in [0, 1]$ to denote the predicted probability of future success of the company represented by $\mathbf{X}_n$, in order to distinguish it from the ground truth label y_n. For the sake of brevity, we use general terms $\mathbf{X}$, y, and $\hat{y}$ to denote $\mathbf{X}_n$, y_n, and $\hat{y}_n$, respectively.

3.1 Time Series Features

We define the input time series features $\mathbf{X}$ by constructing 16 time series that fall into 6 feature categories, as summarized in [9]. These categories are ① **funding**, ② **founder/owner**, ③ **team**, ④ **investor**, ⑤ **web**, and ⑥ **context**, and below we will introduce the selected features under each category. Each time series feature contains precisely T values corresponding to the T time steps. For a concrete example of $\mathbf{X}$, see Fig. 1. All time series features are numerical, with

the exception of the first one (`round_type`), which is categorical. Each time step corresponds to a calendar month, and the steps are aligned monthly.

① **Funding** category contains statistics of historical funding received by the company, showing recognition from investors.

- `round_type` indicates the latest funding round type that a company has received up to time t, such as *Seed* or *Series A*, providing insights into its funding stage and maturity. It is a categorical feature with 60 unique values.
- `total_funding` is the cumulative amount of funding in USD that the company has received up to time t, indicating the amount of capital it has been able to attract. The value range is from 1 to approximately 2×10^{11}.
- `valuation`: the estimated USD valuation of the company immediately after its latest funding round and is included to provide insight into a company's overall financial value. It is a numerical feature with values ranging from 1 to about 1×10^{12}.

② **Founder/Owner**: this category captures attributes of the founding team, which are critical to a company's short-term success and long-term survival [21].

- `n_founder` shows the number of a company's founders still with the company at time t. The value ranges from 0 to 38.

③ **Team**: this category captures the statistics of the employees of the company.

- `n_employee`: the number of employees at time t, implying the company's growth trajectory. The feature has a value range of 1 to 113,757.

④ **Investor** category captures the statistics of investors who have funded the company, indicating its early attractiveness.

- `n_investor` represents the total number of unique investors who have provided funding to the company up to time t. This feature provides insights into the diversification of the company's investment sources. The value ranges from 1 to 240.
- `growth_investor_rate` is the ratio of unique GC investors[8] among the company's unique investors up to time t. This feature indicates the investors' beliefs in the company's future growth potential.
- `average_cagr` is the average Compound Annual Growth Rate (CAGR)[9] of all exited deals made by the company's investors up to time t. This feature is meant to demonstrate the past investment performance of the involved investors.
- `2x_cagr_rate` is calculated as the ratio of investment deals up to time t with a CAGR ≥ 2 among all exited deals made by the company's investors. This feature reflects the proportion of investors with a history of impressive returns who are currently invested in the company.

[8] GC investors are defined as those who have participated in a funding round of 50 million USD or valuation above 200 million USD.

[9] CAGR $= (\mathrm{EV}/\mathrm{SV})^{1/\mathrm{Y}} - 1$ is calculated for each deal the investor has exited, where SV and EV stand for the starting and exiting value of the investment, respectively; Y is the number of holding years (from investment till divestment) of the invested asset.

⑤ **Web**: this category covers any feature extracted from web pages that are related to the company in focus.

- `cu_popularity` describes the company's domain name popularity rank at time t. This rank is determined based on the domain's network traffic as measured by Cisco Umbrella (CU)[10].
- `sw_global_rank` describes at time t the monthly unique visitors and pageviews of the company website(s). The higher this sum, the higher the site's rank. This feature is obtained from SimilarWeb[11].
- `n_desktop_visitor` and `n_mobile_visitor` are two features indicating the number of unique visitors to the company's website utilizing a desktop and mobile device, respectively. Both are sourced from SimilarWeb.
- `n_news` counts the number of times a company is mentioned across approximately 3,700 news websites to a time point t, reflecting its media visibility and recognition. The value range of the dataset is 1 to 389.

⑥ **Context**: this category captures extrinsic factors[12] that may be (but are not limited to) competition, regional, environmental, cultural or economical based.

- `n_regional_seed_round` represents the number of seed funding rounds in the company's region[13] between adjacent time points t–1 and t. This feature offers context on the company's performance relative to regional competitors and financial conditions, highlighting potential company success even if regional investments are low.
- `n_regional_series_ab`: same as the previous one except that it is counting the Series A and B rounds instead.

3.2 TMTSC Architecture

As illustrated in Fig. 2, TMTSC learns to predict $\hat{y}_n$ using time series input $\mathbf{X}$. At the t-th time step, each input feature vector $\mathbf{x}_t$ consists of a numerical part (often normalized), denoted as $\mathbf{u}_t$, and a categorical part, denoted as $\mathbf{v}_t$. Thus, $\mathbf{x}_t = [\mathbf{u}_t; \mathbf{v}_t]$, where ";" represents a vector concatenation operation. To convert the categorical features $\mathbf{v}_t$ into dense embeddings, we utilize embedding layers, which can be collectively represented by a learnable function $\mathcal{E}$. The embedded categorical features are then given by $\mathbf{v}'_t = \mathcal{E}(\mathbf{v}_t)$. As a result, the K-dimensional vector $\mathbf{x}_t$ is transformed to a new numerical vector $\mathbf{x}'_t$ that has K' ($K' > K$) dimensions:

$$\mathbf{x}'_t = [\mathbf{u}_t; \mathcal{E}(\mathbf{v}_t)] \in \mathbb{R}^{K'} \text{ and } \mathbf{x}_t = [\mathbf{u}_t; \mathbf{v}_t] \in \mathbb{R}^{K}. \tag{1}$$

Then, $\mathbf{x}'_t$ is linearly projected onto a D-dimensional vector space, where D is the dimension of the Transformer model sequence element representations:

[10] http://s3-us-west-1.amazonaws.com/umbrella-static/index.html.

[11] https://support.similarweb.com/hc/en-us/articles/213452305-Rank.

[12] While intrinsic features act from within a company, extrinsic ones wield their influence from the outside. The company may impact the former, yet not the latter.

[13] A region is a collection of countries such as *Great Britain*, *DACH*, *France Benelux*, *Southern Europe*, *Nordics*, *South Asia*, *South East Asia*, and so on.

$$\mathbf{h}_t = \mathbf{W}\mathbf{x}'_t + \mathbf{b}, \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{D \times K'}$ and $\mathbf{b} \in \mathbb{R}^D$ are learnable parameters and $\mathbf{h}_t \in \mathbb{R}^D$, $t \in \mathbb{Z} \cap [1, T]$ are the input vectors to the Transformer model. Although Eqs. (1) and (2) show the operation for a single time step for clarity, all raw input vectors $\mathbf{x}_t$, $t \in \mathbb{Z} \cap [1, T]$ are embedded in the same way concurrently. It is worth mentioning that the above formulation can also accommodate univariate time series (i.e., $K = 1$), though in the scope of this work, we will only evaluate the approach on multivariate time series.

It is important to note that the Transformer is a feed-forward architecture that does not inherently account for the order of input elements. To address the sequential nature of time series data, we incorporate positional encodings, denoted as $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T] \in \mathbb{R}^{T \times D}$, to the input vectors $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T] \in \mathbb{R}^{T \times D}$, resulting in the final input $\mathbf{H}'$:

$$\mathbf{H}' = \mathbf{H} + \mathbf{P} = [\mathbf{h}'_1, \mathbf{h}'_2, \ldots, \mathbf{h}'_T] \in \mathbb{R}^{T \times D}, \tag{3}$$

where $\mathbf{h}'_t \in \mathbb{R}^D = \mathbf{h}_t + \mathbf{p}_t$. Closely following the approach in [44], we employ fully learnable positional encodings, as they have been reported to yield better performance compared to deterministic sinusoidal encodings [40] for multivariate time series classification tasks. We also utilize batch normalization (rather than layer normalization), as it is considered effective in mitigating the impact of outlier values in time series data, an issue that does not arise for textual inputs.

The Transformer-based model architecture depicted in Fig. 2 generates T output vectors z_t corresponding to the T input time steps. These output vectors are concatenated to form a single output matrix $\mathbf{Z} = [\mathbf{z}_1; \mathbf{z}_2; \ldots; \mathbf{z}_T]$, which serves as the input for a linear layer. As shown in Eq. (4), the linear layer is parameterized by $\mathbf{W}_{\text{out}} \in \mathbb{R}^{C \times (T \cdot D)}$ and $\mathbf{b}_{\text{out}} \in \mathbb{R}^C$, where C denotes the number of classes to be predicted.

$$\hat{\mathbf{y}} = \mathit{Softmax}(\mathbf{W}_{\text{out}}\mathbf{Z} + \mathbf{b}_{\text{out}}). \tag{4}$$

3.3 Optimization Target

In the absence of a universally agreed-upon definition of "true success" of startups and scaleups, most existing definitions tend to focus on "growth", which can be measured from various perspectives, such as funding, revenue, employee count,

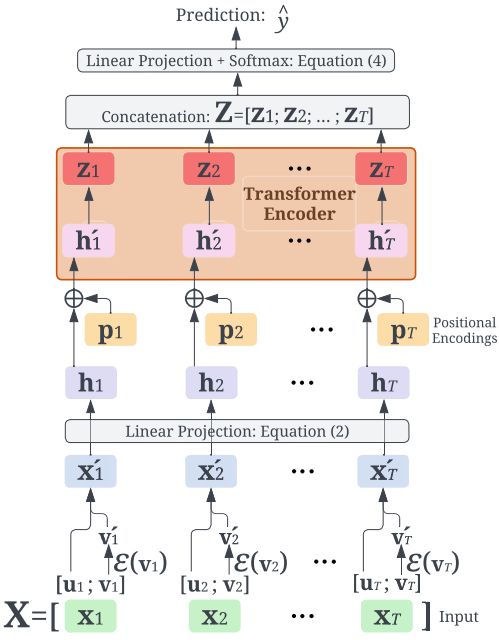


Fig. 2. TMTSC architecture: $\mathbf{u}_t$ and $\mathbf{v}_t$ are numerical and categorical part respectively.

Table 1. Specification of tasks/datasets: split with an investor-centric strategy [9].

Dataset	#feat.	#sample	#time step	#class	#train	#val.	#test
VC	16	86,886	24	2	63,562	11,178	12,146
GC	16	21,163	24	2	16,275	–	4,888

and valuation, among others [9]. As a well-established investment firm, we have access to a large volume of expert evaluations (akin to [4,28]) that represent quantified assessments from human experts. These evaluations encompass multiple categories/terms, such as "inbound", "reviewing", "reach-out", "follow", "negotiating", and "out-of-scope"[14], which are updated periodically by investment professionals for companies in the context of VC and GC. To further simplify the prediction task, we assign each evaluation term to either a good ("`1`") or bad ("`0`") binary bucket denoted by $\mathbf{y}_n$ in Fig. 1, implying $C = 2$ in Eq. (5). In this way, each company is annotated with two ground-truth binary labels – one for VC and the other for GC; and the loss function $\mathcal{L}$ is

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\left[\mathbf{y}_n \log(\hat{\mathbf{y}}_n) + (1-\mathbf{y}_n)\log(1-\hat{\mathbf{y}}_n)\right]. \tag{5}$$

4 Experiments on Real-World Investment Tasks

Following the details introduced in the previous section, we prepare two real-world proprietary datasets: "**VC**" for the VC context and "**GC**" for the GC context, performing data augmentation to obtain monthly time steps. To eliminate overly sparse time series, we discard the samples whose time series features are all shorter than six months. Missing `valuation` values are approximated with the cumulative funding received up to that point. Missing `total_funding` values are filled by taking the value of the previous month (if available) or 0 otherwise. For the time steps where the values are still missing, we fill them with "−1". Finally, we pad all time series to the same length of 24 months. As for scaling, we empirically apply log-transform to 13 numerical features (excluding `cu_popularity` and `n_employee`). The specification is presented in Table 1. It is worth noting that we also experimented with two public TSC (time series classification) benchmark datasets[15]: Ethanol [30] and PEMS-SF [17]. Since they do not directly relate to the investment business domain, we chose to leave them outside the scope of this paper. For in-depth information about experiments on public datasets, we recommend reading [25].

[14] The complete evaluation framework is withheld as it is proprietary.

[15] The overall performance can be found on Motherbrain's blog post: https://motherbrain.ai/applying-transformers-to-score-potentially-successful-startups-7893284efb01.

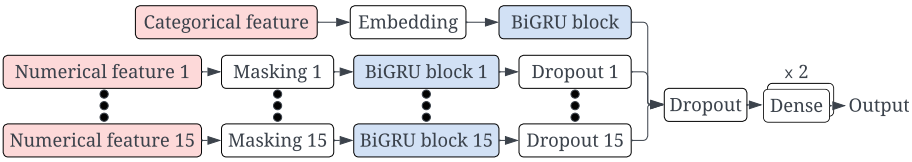


Fig. 3. U-GRU: each univariate time series is modeled by a BiGRU block.

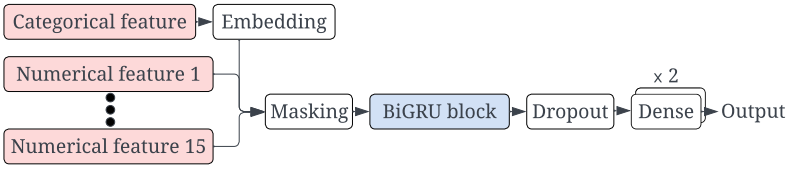


Fig. 4. M-GRU: all time series features are modeled by one single BiGRU block.

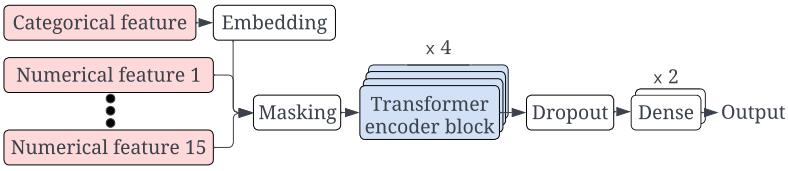


Fig. 5. TE architecture with 4 Transformer encoder blocks.

4.1 Baselines and Hyper-parameters

GRU (Gated Recurrent Unit) [16] is a highly relevant baseline for comparison due to its ability to model sequential dependencies and capture long-term dependencies in time series data. We experiment both U-GRU (Univariate GRU) and M-GRU (Multivariate GRU), whose architectures are illustrated in Figs. 3 and 4 respectively. We adopt the implementation of BiGRU (Bidirectional GRU) [15], masking, embedding, dropout, and dense layers from Keras[16]. In both architectures, the masking layer is added to inform the model to ignore any values marked as missing in its computation. As a close relative and foundation of TMTSC, the Transformer Encoder (TE) [40] is also selected as a baseline. As shown in Fig. 5, we adopt the same layers as the original implementation [40]. For comparability, the input features are ingested, embedded and concatenated in the same way as M-GRU as shown in Fig. 4. The hyper-parameters are selected based on the highest AUC-ROC ("Area Under the Curve" of the Receiver Operating Characteristic curve) score on the validation split of the dataset. Refer to [25] for the searched and selected hyper-parameter values.

[16] Keras Layer documentation: https://keras.io/api/layers.

4.2 Overall Performance: A Precision-Centric Comparison

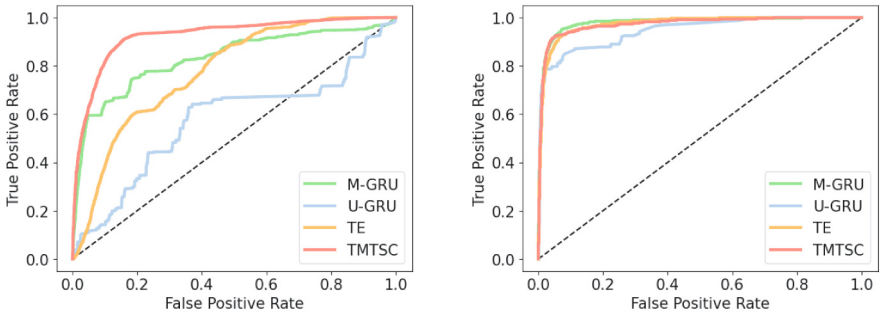
When interpreting the results in Table 2, the costs of different types of prediction errors must be considered. The outcome from false negatives (failing to identify a successful company) is that investors are simply not made aware of a successful company and therefore no action is taken. In that regard, there is an upside loss in terms of lost profit but no detriment in terms of time or money invested. False positives (incorrectly predicting a company will be successful), on the other hand, can lead to wasted time spent on due diligence, or, in the worst case, an investment that loses money. For that reason, it is more important to evaluate a model with respect to its precision, or the number of its positive predictions that are actually positive. Observing the precision scores in Table 2, TMTSC outperforms all other methods, achieving scores of 0.86 and 0.83 for VC and GC scenario, respectively. Additionally, Fig. 6 provides a more balanced and comprehensive view using AUC-ROC metric. TMTSC clearly outperforms on the VC task, achieving an average score of 0.92, 12% better than M-GRU, the next

Table 2. Overall performance comparison.

Task	Metric	U-GRU	M-GRU	TE	TMTSC
VC	Accuracy ± STDEV	0.548 ± **0.013**	0.731 ± 0.026	0.655 ± 0.081	**0.863** ± 0.015
	Precision ± STDEV	0.704 ± **0.015**	0.740 ± 0.044	0.699 ± 0.062	**0.864** ± 0.016
	AUC-ROC ± STDEV	0.628 ± 0.015	0.819 ± 0.020	0.780 ± 0.081	**0.924** ± **0.009**
GC	Accuracy ± STDEV	0.934 ± 0.011	0.924 ± 0.027	0.933 ± 0.021	**0.956** ± **0.004**
	Precision ± STDEV	0.701 ± 0.058	0.794 ± 0.101	0.765 ± 0.108	**0.831** ± **0.026**
	AUC-ROC ± STDEV	**0.977** ± 0.008	0.939 ± 0.002	0.971 ± 0.002	0.971 ± **0.001**

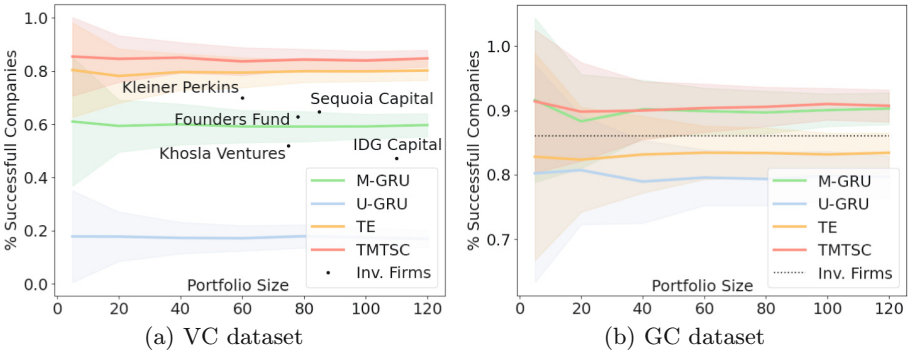
Table 3. The comparison of training efficiency. Underlined values indicate shortest time per step for each dataset.

Task	Method	Sec./Step	Relative Time
VC	U-GRU	2.000	83.3 ×
	M-GRU	<u>0.024</u>	1.0 ×
	TE	0.057	2.4 ×
	TMTSC	0.100	4.2 ×
GC	U-GRU	1.232	46.6 ×
	M-GRU	<u>0.026</u>	1.0 ×
	TE	0.056	2.1 ×
	TMTSC	0.101	3.8 ×



(a) VC dataset (b) GC dataset

Fig. 6. ROC (Receiver Operating Characteristic) curves.



(a) VC dataset (b) GC dataset

Fig. 7. Portfolio simulation: success rate vs. portfolio size.

best method. All methods perform extremely well on GC task, and TMTSC's AUC-ROC score of 0.97 is less than 1% lower than the winner U-GRU.

4.3 Training Stability and Efficiency

The standard deviation values (STDEV) in Table 2 indicate that the TMTSC training is relatively stable in both the VC and GC datasets, as the values are relatively low compared to the other baselines. To measure training efficiency, we record per-step training time for each dataset and method using the same batch size (=512) and hardware configurations. The results are presented in Table 3, where the "Relative Time" column shows how the time consumption for the corresponding method relates to the fastest method (i.e., "1.0 ×"). Take the VC task for example, "2.4 ×" for TE would therefore mean TE took over twice as long as M-GRU. It is evident that M-GRU requires the least amount of training time, largely due to its design, which favors simplicity. TMTSC and TE take only a small amount of extra time to train, despite their increased complexity. This is likely due to their multi-head architecture, allowing parallelization of self-attention computations.

4.4 Portfolio Simulation

To further evaluate the model in the context of the real-world investment scenario, portfolio simulations are executed and visualized in Fig. 7. Concretely, we assemble a set by isolating the companies confirmed to be potentially good investment targets in the VC or GC datasets (i.e., that are positively labeled). From this set, we randomly sample i companies to simulate forming VC/GC investment portfolios of size i and calculate the percentage of companies each model predicts to be successful within the sample. To address the stochasticity of this process, we perform each simulation 100 times. Different portfolio sizes (i.e., values of i) are simulated; and for each i (X-axis), the mean and standard deviations are plotted (Y-axis), resulting in a colored line with a shaded area in

Fig. 7. In VC and GC contexts, we can see that (1) TMTSC performs the best among all methods, (2) performance becomes less variable as simulated portfolio size increases, and (3) the models evaluated performed more variably on the VC dataset than the GC dataset.

To roughly compare these methods against real-world VC and GC fund performance, Fig. 7(a), includes the portfolio size and performance of five VC funds [42], showing a performance largely on par with M-GRU and inferior to TMTSC and TE. For the GC simulation, a horizontal line representing the real-world GC success rate of 86.3% [33] is included in Fig. 7(a). Here, the real-world GC success rate is outperformed by M-GRU and TMTSC. It is important to note that investment firms are much more constrained than the simulation: they cannot invest in every attractive company they encounter due to factors like founders' preference, portfolio conflict, investment focus, and available funds.

5 Conclusion and Future Work

In this work, we propose using a Transformer-based Multivariate Time Series Classifier (TMTSC) to facilitate sourcing investment targets for Venture Capital (VC) and Growth Capital (GC). Specifically, TMTSC utilizes multivariate time series as input to predict the probability that any candidate company will succeed in the context of a VC or GC fund. We formally define the sourcing problem as a multivariate time series classification task, and introduce the key components of our implementation, including input features, model architecture, and optimization target. Our extensive experiments on two proprietary datasets (collected from real-world VC and GC contexts) demonstrate the effectiveness, stability, and efficiency of our approach compared with three popular baselines. To further evaluate the model in the context of the real-world investment scenario, portfolio simulations are executed, showing TMTSC's high success rate in both VC and GC sourcing. The main future work includes (1) incorporating global features along with time series input, (2) and learning generic and condensed representations for multivariate time series for varies downstream prediction tasks.

References

1. Allu, R., Padmanabhuni, V.N.R.: Predicting the success rate of a start-up using LSTM with a swish activation function. J. Control Decis. **9**(3), 355–363 (2022)
2. Ang, Y.Q., Chia, A., Saghafian, S.: Using machine learning to demystify startups' funding, post-money valuation, and success. In: Innovative Technology at the Interface of Finance and Operations, pp. 271–296. Springer, Cham (2022)
3. Arroyo, J., Corea, F., Jimenez-Diaz, G., Recio-Garcia, J.A.: Assessment of machine learning performance for decision support in venture capital investments. IEEE Access **7**, 124233–124243 (2019)
4. Bai, S., Zhao, Y.: Startup investment decision support: application of venture capital scorecards using machine learning approaches. Systems **9**(3), 55 (2021)

5. Blank, S.: Why the lean start-up changes everything. Harvard Bus. Rev. **91**(5), 63–72 (2013)
6. Block, J., Fisch, C., Vismara, S., Andres, R.: PE investment criteria: an experimental conjoint analysis of venture capital, business angels, and family offices. J. Corp. Finan. **58**, 329–352 (2019)
7. Bonaventura, M., Ciotti, V., Panzarasa, P., Liverani, S., Lacasa, L., Latora, V.: Predicting success in the worldwide start-up network. Sci. Rep. **10**(1), 1–6 (2020)
8. Cao, L., et al.: A scalable and adaptive system to infer the industry sectors of companies: prompt + model tuning of generative language models. In: Proceedings of the IJCAI Workshop on Financial Technology and Natural Language Processing (FinNLP), pp. 1–11, August 2023
9. Cao, L., von Ehrenheim, V., Krakowski, S., Li, X., Lutz, A.: Using deep learning to find the next unicorn: a practical synthesis. arXiv preprint arXiv:2210.14195 (2022)
10. Cao, L., Horn, S., von Ehrenheim, V., Stahl, R.A., Landgren, H.: Simulation-informed revenue extrapolation with confidence estimate for scaleup companies using scarce time series data. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM 2022), 17–21 October 2022, Atlanta, GA, USA, p. 12. Association for Computing Machinery (ACM), New York, NY, USA, October 2022
11. Cavallo, A., Ghezzi, A., Dell'Era, C., Pellizzoni, E.: Fostering digital entrepreneurship from startup to scaleup: the role of venture capital funds and angel groups. Technol. Forecast. Soc. Chang. **145**, 24–35 (2019)
12. Chen, M., et al.: A trend-aware investment target recommendation system with heterogeneous graph. In: International Joint Conference on Neural Networks, pp. 1–8 (2021)
13. Cheng, C., Tan, F., Hou, X., Wei, Z.: Success prediction on crowdfunding with multimodal deep learning. In: IJCAI, pp. 2158–2164 (2019)
14. Chernenko, S., Lerner, J., Zeng, Y.: Mutual funds as venture capitalists? Evidence from unicorns. Rev. Finan. Stud. **34**(5), 2362–2410 (2021)
15. Cho, K., et al.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar, October 2014
16. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS 2014 Workshop on Deep Learning, December 2014 (2014)
17. Cuturi, M.: Fast global alignment kernels. In: Proceedings of the 28th International Conference on Machine Learning (ICML-2011), pp. 929–936 (2011)
18. Dellermann, D., Lipusch, N., Ebel, P., Popp, K.M., Leimeister, J.M.: Finding the unicorn: predicting early stage startup success through a hybrid intelligence method. In: International Conference on Information Systems (2021)
19. Garkavenko, M., Gaussier, E., Mirisaee, H., Lagnier, C., Guerraz, A.: Where do you want to invest? Predicting startup funding from freely, publicly available web info. arXiv:2204.06479 (2022)
20. Gastaud, C., Carniel, T., Dalle, J.M.: The varying importance of extrinsic factors in the success of startup fundraising: competition at early-stage and networks at growth-stage. arXiv:1906.03210 (2019)
21. Ghassemi, M., Song, C., Alhanai, T.: The automated venture capitalist: data and methods to predict the fate of startup ventures. In: AAAI Workshop on Knowledge Discovery from Unstructured Data in Financial Services (2020)

22. Gompers, P.A., Gornall, W., Kaplan, S.N., Strebulaev, I.A.: How do venture capitalists make decisions? J. Financ. Econ. **135**(1), 169–190 (2020)
23. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
24. Guerzoni, M., Nava, C.R., Nuccio, M.: The survival of start-ups in time of crisis: a ML approach to measure innovation. arXiv preprint arXiv:1911.01073 (2019)
25. Halvardsson, G.: A Transformer-based scoring approach for startup success prediction. Master's thesis, KTH Royal Institute of Technology (2023)
26. Horn, S.: Deep learning models as decision support in venture capital investments: temporal representations in employee growth forecasting of startup companies. Master's thesis, KTH Royal Institute of Technology & EQT Partners (2021)
27. Kaiser, U., Kuhn, J.M.: The value of publicly available, textual and non-textual information for startup performance prediction. J. Bus. Ventur. Insights **14**, e00179 (2020)
28. Kinne, J., Lenz, D.: Predicting innovative firms using web mining and deep learning. PLoS ONE **16**(4), e0249071 (2021)
29. Krishna, A., Agrawal, A., Choudhary, A.: Predicting the outcome of startups: less failure, more success. In: International Conference on Data Mining Workshop, pp. 798–805 (2016)
30. Large, J., Kemsley, E.K., Wellner, N., Goodall, I., Bagnall, A.: Detecting forged alcohol non-invasively through vibrational spectroscopy and machine learning. In: Phung, D., Tseng, V.S., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (eds.) PAKDD 2018. LNCS (LNAI), vol. 10937, pp. 298–309. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93034-3_24
31. Lyu, S., et al.: Graph neural network based VC investment success prediction. arXiv preprint arXiv:2105.11537 (2021)
32. Malmström, M., Voitkane, A., Johansson, J., Wincent, J.: What do they think and what do they say? Gender bias, entrepreneurial attitude in writing and venture capitalists' funding decisions. J. Bus. Ventur. Insights **13**, e00154 (2020)
33. Mooradian, P., Auerback, A., Slotsky, C., Gilfix, J.: Growth equity: turns out, it's all about the growth (2019)
34. Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., Sohl-Dickstein, J.: On the expressive power of deep neural networks. In: International Conference on Machine Learning, pp. 2847–2854. PMLR (2017)
35. Santisteban, J., Mauricio, D., Cachay, O., et al.: Critical success factors for technology-based startups. Int. J. Entrep. Small Bus. **42**(4), 397–421 (2021)
36. Shi, J., Yang, K., Xu, W., Wang, M.: Leveraging deep learning with audio analytics to predict the success of crowdfunding projects. J. Supercomput. **77**(7), 7833–7853 (2021)
37. Skawińska, E., Zalewski, R.I.: Success factors of startups in the EU - a comparative study. Sustainability **12**(19), 8200 (2020)
38. Stahl, R.H.A.: Leveraging time-series signals for multi-stage startup success prediction. Master's thesis, ETH Zurich & EQT Partners (2021)
39. Tang, Z., Yang, Y., Li, W., Lian, D., Duan, L.: Deep cross-attention network for crowdfunding success prediction. IEEE Trans. Multimedia **25**, 1306–1319 (2022)
40. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
41. Williamson, K.: Research Methods for Students, Academics and Professionals: Information Management and Systems. Elsevier (2002)

42. Yin, D., Li, J., Wu, G.: Solving the data sparsity problem in predicting the success of the startups with machine learning methods. arXiv preprint arXiv:2112.07985 (2021). https://arxiv.org/abs/2112.07985
43. Żbikowski, K., Antosiuk, P.: A machine learning, bias-free approach for predicting business success using Crunchbase data. Inf. Process. Manage. **58**(4), 102555 (2021)
44. Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C.: A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2114–2124 (2021)
45. Zhang, S., Zhong, H., Yuan, Z., Xiong, H.: Scalable heterogeneous graph neural networks for predicting high-potential early-stage startups. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2202–2211 (2021)