DIFFSTOCK: PROBABILISTIC RELATIONAL STOCK MARKET PREDICTIONS USING DIFFUSION MODELS

Divyanshu Daiya[1], *Monika Yadav*[2], *Harshit Singh Rao*[3]

[1]Department of Computer Science, Purdue University
[2]Thomas Lord Department of Computer Science, University of Southern California, [3]NexusQuant AI
[1]divyanshu@purdue.edu, [2]monikaya@usc.edu, [3]harshitrao@nexusquant.tech

ABSTRACT

In this work, we propose an approach to generalize denoising diffusion probabilistic models for stock market predictions and portfolio management. Present works have demonstrated the efficacy of modeling interstock relations for market time-series forecasting and utilized Graph-based learning models for value prediction and portfolio management. Though convincing, these deterministic approaches still fall short of handling uncertainties i.e., due to the low signal-to-noise ratio of the financial data, it is quite challenging to learn effective deterministic models. Since the probabilistic methods have shown to effectively emulate higher uncertainties for time-series predictions. To this end, we showcase effective utilisation of Denoising Diffusion Probabilistic Models (DDPM), to develop an architecture for providing better market predictions conditioned on the historical financial indicators and inter-stock relations. Additionally, we also provide a novel deterministic architecture MaTCHS which uses Masked Relational Transformer(MRT) to exploit inter-stock relations along with historical stock features. We demonstrate that our model achieves SOTA performance for movement predication and Portfolio management.

Index Terms— Diffusion Models, Stock Market, Relational Learning

1. INTRODUCTION

Stock price prediction is an age-old intrigue for investors due to its potential dividends and the inherent challenges it presents due to market volatility and its stochastic nature [1, 2]. Modern advancements in Deep Learning now empower researchers to employ a multitude of modalities such as historical stock trends, news, social media, and financial reports in market prediction models [3, 4]. There has been a concerted effort in modeling inter-stock dependencies, revealing that stocks associated with top-tier positions, or those in similar sectors, often show correlated trends, contributing to significant improvements in market predictions [5, 6].

Historically, time series prediction relied heavily on state space framework-based statistical models such as ARIMA and exponential smoothing [7, 8]. However, while pure machine learning approaches have been explored, they often didn't surpass statistical models due to issues like overfitting and non-stationarity [9, 10]. In recent years, the spotlight has shifted towards the diffusion model for probabilistic time series forecasting, marking state-of-the-art performances. Notably, the TimeGrad model [11] and CSDI model [12] have showcased the potency of the diffusion model for optimizing forecasting.

Yet, a challenge persists in the realm of stock prediction: the low signal-to-noise ratio inherent in stock data. Such noise can hinder machine learning models' effectiveness, affecting the accuracy of latent factors [13]. While the integration of multi-modal data helps bridge the gap, deterministic methods often grapple with the uncertainties introduced over time. This has led to the rise of probabilistic models, notably the Denoising Diffusion Probabilistic Models (DDPM) [14], which transform noise into predictions using a denoising process conditioned on historical readings. However, their limitation lies in solely modeling temporal dependencies, neglecting spatial correlations between stock nodes [11, 12].

To surmount these challenges, we propose a novel framework that synergizes DDPMs with relational market data, encapsulating the spatio-temporal strengths of deterministic models and the uncertainty handling of DDPMs. Our MaTCHS architecture, derived from our preceding work [15], employs Transformer Encoders with Masked attention heads, encapsulating both temporal dynamics and spatial correlations.

For a detailed exploration of deterministic models, readers are encouraged to refer to [15, 16].

2. MODEL

2.1. General Market Prediction Task

We are interested in the problem of predicting future stock prices, and we are provided with P financial indicators for the last L days for every stock. We have a total of N stocks, and which gives a set of financial and social indicators represented as, $f_{N,L} = (f'^{1}_{P,L}, f'^{2}_{P,L} ..., f'^{N}_{P,L})$ here $f'^{m}_{P,L} \in \mathbb{R}^{P \times L}$. In addition, we have a relation matrix $C \in \mathbb{R}^{N \times N \times G}$, which specifies connections between N stocks over G different relations, $C(i, j, k) = \{1$ if i and j are connected by relation

k, else 0}[6]. We can model our relation matrix as a graph $H = (\mathcal{N}, \mathcal{E})$, with $\mathcal{E} \in \mathbb{R}^G$ is a hyperedge.
Given historical financial signals $f_{N,L}$ we can represent each node as a stock in the graph H. Now, given this information we seek to predict the stock value at time step $T_{observed} + 1$ to T for all the N stocks, i.e. $x^{(T+1,N)}$. So, we can formulate our problem as $p(x^{(T+1,N)}|f_{N,L}, H)$ and we seek to learn some $F : (f_{N,L}, H) \rightarrow x^{(T+1,N)}$.

2.2. Conditional Diffusion Model

We build on the work by [12, 17], to develop a conditional diffusion model for market prediction. We can start by following a obvious approach using the stock trend history $f_{t,L}$ and relational data between stocks H as the condition in the reverse process. So, we can write our conditioned reverse diffusion process as,

$$p_\theta(x_{0:K}^{(T+1,N)}|f_{N,L}, H) = p(x_K^{(T+1,N)}) \times \prod_{k=K}^{1} p_\theta(x_{k-1}^{(T+1,N)}|x_k^{(T+1,N)}, f_{N,L}, H). \quad (1)$$

Here we can note that $x^{(T+1,N)}$ can be thought of as being sampled from the same distribution as $f_{N,L}$ due to the high trend correlation between consecutive time-stamps. To better utilize this association, we modify the equation to predict $f_{N,L+1}$ i.e. $f_{N,L}$ along with future time steps as demonstrated by [17]. This formulation provides a unifying approach that combines historical reconstruction and future estimation. By predicting $f_{N,L+1}$, we can use historical data to model the distribution of data comprehensively.

$$p_\theta((f_{N,L+1})_{0:K}|f_{N,L}, H) = p((f_{N,L+1})_K) \times \prod_{k=K}^{1} p_\theta((f_{N,L+1})_{k-1}|(f_{N,L+1})_k, f_{N,L}, H). \quad (2)$$

with training objective,

$$\min_\theta \mathcal{L}(\theta) = \min_\theta \mathbb{E}_{(f_{N,L+1})_0,\epsilon} \left\| \epsilon - \epsilon_\theta \left((f_{N,L+1})_k, k | f_{N,L}, H \right) \right\|_2^2 . \quad (3)$$

Here, ϵ_θ is our MaTCHS model, the denoising model. The denoising function ϵ_θ estimates the noise vector ϵ that was added to its noisy input $(f_{N,L+1})_k$. Detailed formulation is similar to works of [11, 12, 14]. [1]

2.3. Adaptive Noise for Financial Series Diffusion

Stock market dynamics are inherently stochastic and diverse, exhibiting various unique patterns like spikes, trends, and declines at different time points. To adequately model and capture these diverse dynamics, it's imperative for the Difffusion process to model inherent volatility at different time points and the collective behaviors of groups of stocks.

Modeling Time Series Variance: One way to assess the intrinsic unpredictability or volatility of the stock market is by evaluating the local variance at each time point. This local variance can be thought of as an indicator of how sensitive a stock might be to broader market fluctuations. The formula is given by, $v(t) = \frac{\sum_{i=t-w}^{t+w} (f_{N,t} - f_{N,i})^2}{2w+1}$, Where the normalization of this variance is represented as, $v_{norm}(t) = \frac{v(t)}{\max_{\tau \in T} v(\tau)}$.

Modeling Intra-Cluster Dynamics: Groups of stocks often exhibit collective behaviors, especially if they belong to the same sector or are influenced by similar macroeconomic factors. To model such collective behaviors or intra-cluster dynamics, we employ Dynamic Time Warping (DTW)[18] method. DTW provides a measure of similarity between time series of individual stocks. Thus, the intra-cluster DTW distance for a stock within a cluster can be expressed as:

$$DTW_{\text{intra}}(f_{N,i}, C) = \text{DTW}\left(\frac{1}{|C| - 1} \sum_{f_{N,j} \in C, i \neq j} f_{N,i}, f_{N,j} \right)$$

The influence of a stock's time series in relation to its cluster is then given by, $I_{intra}(f_{N,i}, C) = \frac{1}{1 + DTW_{intra}(f_{N,i}, C)}$.

Integrating Volatility and Cluster Dynamics: To obtain a comprehensive understanding of a stock's behavior, both its individual volatility and its intra-cluster dynamics should be taken into account. By integrating these, we get:

$$v_{score,intra}(t, f_{N,i}, C) = \alpha \times v_{norm}(t) + (1 - \alpha) \times I_{intra}(f_{N,i}, C) \times v_{norm}(t, C),$$

This integrated score provides a unified metric, represented as $I(t) = v_{score,intra_{norm}}(t, C)$, that captures the overall significance of a given time point. Based on this metric, noise can be adaptively applied to market signal :

$$q((f_{N,L+1})_{k+1}|(f_{N,L+1})_k) = N((f_{N,L+1})_{k+1}; \sqrt{(1 - \beta_t)}(f_{N,L+1})_k, \beta_t I). \quad (4)$$

This ensures that significant time points and relation patterns are emphasized during Diffusion process by prioritizing learning these trends during denoising process and enabling model in making better future trend prediction.

2.4. MaTCHS(Denoising Model)

Introducing **MaTCHS**, our architecture fuses a **Ma**sked **T**ransformer with a **C**onvoutional network to predict stock prices leveraging **H**ypergraph relations. It consists of two main segments. The first, Att-DiCEm, focuses on temporal feature extraction from financial indicators using the Att-DCNN approach[16]. The second emphasizes understanding the relationship between different stocks for price prediction.

[1] It is to be noted that at all the instance where we have used $T + 1$ and $L + 1$ it can be generalised to $T + t'$ and $L + l'$ i.e. any number of future time steps t' or l' can be predicted.

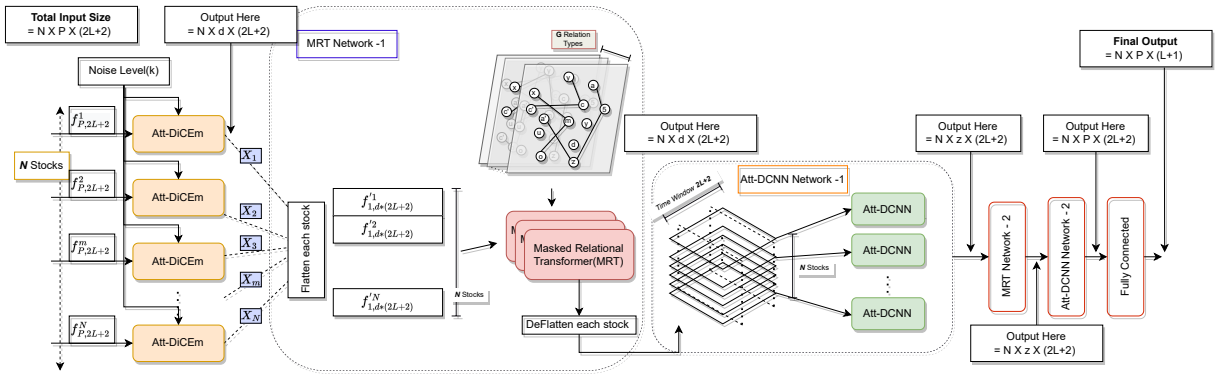

Fig. 1. $MaTCHS$ Denoising Model: Masked Transformer and Convoutional network for Hypergraph relation based Stock time-series generation.

After the "Masked Relational Transformer(MRT)" provides relational information, this is further enhanced temporally with another Att-DCNN layer. The model concludes by producing a financial time-series for each stock spanning past and future time-stamps. The input to our denoising network will be concatenated $(f_{N,L+1})_k$, and $f_{N,L+1}$, the conditional for the diffusion model with mask at the future time-steps. So the concatenated input takes the form of $f_{N,2(L+1)}$.

Att-DiCEm Building upon our previous work[15], the event pipeline has been excluded, emphasizing the financial segment. We've adapted this segment to generate a time series across 2L+1 time steps with "d" time variables. Contrasting the original output of $\mathbb{R}^{(2L+1)\times 1}$, the new format is $\mathbb{R}^{(2L+1)\times d}$. Noise level positional encoding, $k \in [1, K]$, is implemented using a transformer positional embedding[19] as, $\mathbf{ne}(k) = [\dots, \cos(k/r^{-2s/D}), \sin(k/r^{-2s/D}), \dots]^{\mathrm{T}}$, with the embeddings added post the initial DC-CNN layer in each Att-DiCEm unit.

2.4.1. Masked Relational Transformer (MRT)

As described in [19], in the Self-Attention Network an attention function maps a query and a set of key-value pairs to an output as, $\mathcal{A}(Q, K, V) = \mathcal{S}(Q, K)V$, with $\mathcal{S}(Q, K) = \left[\frac{\exp\left(Q_i K_j^T/\sqrt{d_k}\right)}{\sum_k \exp\left(Q_i K_k^T/\sqrt{d_k}\right)}\right]$. Here, queries Q, keys K, and values $V \in \mathbb{R}^{T\times d_k}$ are matrices. A representation sequence is given as $H^l \in \mathbb{R}^{T\times d}$ in the l-th layer, $H^l = \left[A^1, \cdots, A^I\right]W_H$, i denotes the attention head and d is the hidden size.

By analyzing the self-attention function, we can model it as a graph. The relation from Node j to i can be described by, $rel(\text{i to j}) = q_i^{\mathrm{T}} k_j$. A modified attention function, incorporating masking to make the model operate on specific graph nodes, is described as, $\mathcal{A}_M(Q, K, V) = S_M(Q, K)V$ with, $\mathcal{S}_M(Q, K) = \left[\frac{M_{i,j}\exp\left(Q_i K_j^T/\sqrt{d_k}\right)}{\sum_k M_{i,k}\exp\left(Q_i K_k^T/\sqrt{d_k}\right)}\right]$, where $M \in \mathbb{R}^{T\times T}$, $M_{i,j} \in [0, 1]$ can be a static or dynamic mask matrix.

Now to model hyper-graph structure one approach would be of aggregating all relation types into a single matrix, but that might overlook specific relation nuances [19]. Given that multi-head attention mechanisms sometimes capture redundant features [20, 21, 22], we propose the Masked Relational Attention Networks (MRAN). It uses separate attention heads for each relation, $\mathcal{A}^i_{M^i}(Q^i, K^i, V^i) = \mathcal{S}^i_{M^i}(Q^i, K^i)V^i$ with, $\mathcal{S}^i_{M^i}(Q^i, K^i) = \left[\frac{M^i_{i,j}\exp\left(Q^i_i K^{iT}_j/\sqrt{d_k}\right)}{\sum_k M^i_{i,k}\exp\left(Q^i_i K^{iT}_k/\sqrt{d_k}\right)}\right]$, such that, A^i_M is Attention calculated for i^{th} attention head which uses i^{th} relation mask M^i i.e. $C(:, :, i)$.

Due to computational concerns, we restrict attention heads in our transformer to 12, grouping similar relations together and using aggregated relations as masks. So, Masked Relational Transformer (MRT), divides the total G relations among 12 attention heads. We also employ additional 4 unmasked attention heads for capturing potential evolving relations between stocks. So, we employ grouped stock-relations C as our second conditional. Given output $X = X_1, \dots, X_N$ from the Att-DiCEm layer, we first flatten it to $X'_m \in \mathbb{R}^{d(2L+1)\times 1}$. MRT uses X'_t as input for Transformer Encoder Layer. The output, X''_t, is obtained by de-flattening each stock in $\mathrm{X}''_{\mathrm{t}}$ to match MRT layer input dimensions. Note that these Transformer Encoders model spatial relations among stocks, not temporal domains.

For training, we retained preprocessing from [15] and used the diffusion model hyperparameters outlined by [12]. We varied β_k capped at $[0, 1, 0.2]$ and diffusion steps K from [20, 50, 100, 200]. Using a batch size of 16 and a decaying learning rate initialized at $1e-4$, training spanned 100 epochs. Two transformer encoders in each MRT Network were used with a prediction window of 3. Evaluation for StockNet was centered on the L+1 timestamp. Training with 200 diffusion steps for 100 epochs on the A100 GPU took 6 hours. Optimal values on StockNet were $K = 100$ and $\beta_k = 0.2$.

3. EVALUATION

3.1. Datasets and Baselines

For an extensive evaluation of our model, we plan to evaluate it on four datasets from US Stock Market spanning over 6 years. We test on NASDAQ [6], NYSE [6] and StockNet [23]. We follow the same train test partitions as in original works [6, 23]. We follow the approach described by [6] for populating relation matrix C for all the datasets. We compare our model with top performing models, STHGCN [24], GCN [25], GCN20 [26], RSR [6], HATS [26]. For Stock Movement Prediction, we'll use accuracy and MCC. For Portfolio Management, we'll compare our cumulative return (IRR) with other trading models and use Sharpe Ratio to measure risk-adjusted returns. For probabilistic models (DDPM), we'll use CRPS to assess distribution compatibility. For portfolio management using predicted prices we will adopt a daily buy-hold-sell strategy as described by [5].

Table 1. Evaluation Results over StockNet Dataset[23]

	Model	$F1\uparrow$	Accuracy $\uparrow$	MCC $\uparrow$
	RAND	$0.502 \pm 8e-4$	$0.509 \pm 8e-4$	$-0.002 \pm 1e-3$
TA	ARIMA - [27]	$0.513 \pm 1e-3$	$0.514 \pm 1e-3$	$-0.021 \pm 2e-3$
	- [28]	$0.529 \pm 5e-2$	$0.530 \pm 5e-2$	$-0.004 \pm 7e-2$
	RandForest - [29]	$0.527 \pm 2e-3$	$0.531 \pm 2e-3$	$0.013 \pm 4e-3$
	TSLDA - [30]	$0.539 \pm 6e-3$	$0.541 \pm 6e-3$	$0.065 \pm 7e-3$
	HAN - [31]	$0.572 \pm 4e-3$	$0.576 \pm 4e-3$	$0.052 \pm 5e-3$
	StockNet - TechnicalAnalyst - [23]	$0.546 \pm -$	$0.550 \pm -$	$0.017 \pm -$
	StockNet - FundamentalAnalyst - [23]	$0.572 \pm -$	$0.582 \pm -$	$0.072 \pm -$
	StockNet - IndependentAnalyst - [23]	$0.573 \pm -$	$0.575 \pm -$	$0.037 \pm -$
FA	StockNet - DiscriminativeAnalyst - [23]	$0.559 \pm -$	$0.562 \pm -$	$0.056 \pm -$
	StockNet - HedgeFundAnalyst - [23]	$0.575 \pm -$	$0.582 \pm -$	$0.081 \pm -$
	GCN[25]	$0.530 \pm 7e-3$	$0.532 \pm 7e-3$	$0.093 \pm 9e-3$
	HATS - [26]	$0.560 \pm 2e-3$	$0.562 \pm 2e-3$	$0.117 \pm 6e-3$
	Adversarial LSTM - [6]	$0.570 \pm -$	$0.572 \pm -$	$0.148 \pm -$
	MAN-SF - [32]	$0.605 \pm 2e-4$	$0.608 \pm 2e-4$	$0.195 \pm 6e-4$
	STHGCN - [5]	$0.609 \pm 2e-4$	$0.613 \pm 2e-4$	$0.198 \pm 6e-4$
	MaTCHS (This work) - AttDiCEm i.e. without relations	$\mathbf{0.568 \pm 2e-3}$	$\mathbf{0.572 \pm 2e-3}$	$\mathbf{0.168 \pm 6e-3}$
	MaTCHS (This work) - Aggregated relations	$\mathbf{0.585 \pm 2e-3}$	$\mathbf{0.587 \pm 2e-3}$	$\mathbf{0.175 \pm 6e-3}$
	MaTCHS (This work)	$\mathbf{0.611 \pm 2e-3}$	$\mathbf{0.612 \pm 2e-3}$	$\mathbf{0.206 \pm 6e-3}$
	MaTCHS (with Diffusion w/o Adap. Noise)	$\mathbf{0.623 \pm 2e-3}$	$\mathbf{0.621 \pm 2e-3}$	$\mathbf{0.214 \pm 6e-3}$
	MaTCHS (with Diffusion)	$\mathbf{0.631 \pm 2e-3}$	$\mathbf{0.634 \pm 2e-3}$	$\mathbf{0.225 \pm 6e-3}$

Table 2. Evaluation Results over NASDAQ and NYSE Dataset(2 decimal places disp.) [6]

Model	NYSE		NASDAQ	
	SR@5	IRR@5	SR@5	IRR@5
ARIMA [27]	$0.33 \pm 3e^{-3}$	$0.10 \pm 5e^{-3}$	$0.55 \pm 1e^{-3}$	$0.10 \pm 6e^{-3}$
A-LSTM [33]	$0.81 \pm 4e^{-3}$	$0.14 \pm 7e^{-3}$	$0.97 \pm 5e^{-3}$	$0.23 \pm 3e^{-3}$
GCN [25]	$0.70 \pm 3e^{-3}$	$0.10 \pm 6e^{-3}$	$0.75 \pm 4e^{-3}$	$0.13 \pm 1e^{-3}$
HATS [26]	$0.73 \pm 5e^{-3}$	$0.12 \pm 2e^{-3}$	$0.80 \pm 6e^{-3}$	$0.15 \pm 7e^{-3}$
DQN [34]	$0.72 \pm 5e^{-3}$	$0.12 \pm 4e^{-3}$	$0.93 \pm 5e^{-3}$	$0.20 \pm 6e^{-3}$
iRDPG [35]	$0.85 \pm 7e^{-3}$	$0.18 \pm 3e^{-3}$	$1.32 \pm 5e^{-3}$	$0.28 \pm 4e^{-3}$
Rank LSTM [36]	$0.79 \pm 1e^{-3}$	$0.12 \pm 6e^{-3}$	$0.95 \pm 4e^{-3}$	$0.22 \pm 2e^{-3}$
GCN [25]	$0.72 \pm 7e^{-3}$	$0.16 \pm 3e^{-3}$	$0.46 \pm 4e^{-3}$	$0.13 \pm 5e^{-3}$
RSR-E [6]	$0.88 \pm 6e^{-3}$	$0.20 \pm 3e^{-3}$	$1.12 \pm 5e^{-3}$	$0.26 \pm 4e^{-3}$
RSR-I [6]	$0.95 \pm 1e^{-3}$	$0.21 \pm 3e^{-3}$	$1.34 \pm 6e^{-3}$	$0.39 \pm 5e^{-3}$
STHAN-SR [5]	$1.10 \pm \pm e^{-3}$	$0.255 \pm e^{-3}$	$1.40 \pm 7e^{-3}$	$0.44 \pm 1e^{-2}$
MaTCHS	$\mathbf{1.13} \pm \pm e^{-3}$	$\mathbf{0.267} \pm e^{-3}$	$\mathbf{1.45} \pm 7e^{-3}$	$\mathbf{0.45} \pm 1e^{-2}$
MaTCHS(Agg)	$\mathbf{0.97} \pm \pm e^{-3}$	$\mathbf{0.22}1 \pm e^{-3}$	$\mathbf{1.34} \pm 7e^{-3}$	$\mathbf{0.40} \pm 1e^{-2}$
MaTCHS(16)	$\mathbf{1.14} \pm \pm e^{-3}$	$\mathbf{0.27}0 \pm e^{-3}$	$\mathbf{1.46} \pm 7e^{-3}$	$\mathbf{0.46} \pm 1e^{-2}$
MaTCHS with Diffusion w/o Adap. Noise	$\mathbf{1.15} \pm \pm e^{-3}$	$\mathbf{0.27}4 \pm e^{-3}$	$\mathbf{1.48} \pm 7e^{-3}$	$\mathbf{0.46} \pm 1e^{-2}$
MaTCHS with Diffusion	$\mathbf{1.18} \pm \pm e^{-3}$	$\mathbf{0.28}5 \pm e^{-3}$	$\mathbf{1.52} \pm 7e^{-3}$	$\mathbf{0.48} \pm 1e^{-2}$
% Improv. (SOTA w.r.t. STHAN-SR)	7.92	9.81	6.18	8.07

Table 3. Comparison with other Diffusion Models

Diffusion Model	StockNet			
	F1	Accuracy	MCC	CRPS
CSDI [12]	$\mathbf{0.582 \pm 2e-3}$	$\mathbf{0.586 \pm 2e-3}$	$\mathbf{0.170 \pm 6e-3}$	0.092
TimeGrad [11]	$\mathbf{0.596 \pm 2e-3}$	$\mathbf{0.598 \pm 2e-3}$	$\mathbf{0.177 \pm 6e-3}$	0.076
MaTCHS with Diffusion (ours)	$\mathbf{0.631 \pm 2e-3}$	$\mathbf{0.634 \pm 2e-3}$	$\mathbf{0.225 \pm 6e-3}$	0.049

3.2. Results and Analysis

Two variations of our model were evaluated: the Diffusion-based MaTCHS and the naive MaTCHS, which omits diffusion. The naive MaTCHS uses an input size of $N \times P \times L$ for N stocks, P indicators, and L timesteps, with an adjusted output layer ($N \times 1$) for next-day stock predictions. Comparative results are presented in Table-1 and Table-2.

The Diffusion-based MaTCHS excels on the StockNet Dataset[23], outperforming all other models in F1, accuracy, and MCC metrics, and on NASDAQ and NYSE dataset[6] outperforming others on SR and IRR . Without the diffusion component, MaTCHS still performs admirably, matching the STHAN-SR model's performance. This emphasizes the strength of our Masked Relational Transformers (MRT) in grasping complex inter-stock dynamics over other GNN-based techniques like HyperGraph-structured STHAN-SR. Separating temporal and spatial predictions has proved beneficial, addressing the complexities arising from concurrent modeling. This separation fosters precision and leads to superior price trend forecasting.

Our Diffusion architectures' performance underscores our hypothesis on the diffusion models ability in capturing stock market nuances better due to their probabilistic nature. Further, our specialized noise schedule for Diffusion enhances performance over standard Diffusion noise schemes across all Datasets. Our guidance in Diffusion emphasizes learning volatile and relational trends, hence augmenting the denoising models' capability.

Further NASDAQ and NYSE datasets, our models reveal potent Portfolio Return trends. Notably, the Sharpe ratios of 7.92% and 6.18% suggesting considerable advances over previous models for utilisation for automated trading capablities. Also, higher IRR scores signify the model's ability to incorporate distant temporal dynamics in prediction, as higher IRR indicates better annual returns relative to the amount invested.

We also trained and tested CSDI[12] and TimeGrad[11] on StockNet(Table-3), we noted predictable performance declines. As CSDI focuses on time-series imputation, and TimeGrad's diffusion-based training isn't tailored to our goals. Our CRPS scores outperformed both, proving our diffusion model's superiority in capturing data distribution.

An ablation study, examining the impact of aggregating relations across attention heads, exhibited performance drops across the StockNet, NASDAQ, and NYSE datasets. Amplifying the attention heads number yields marginal improvements, reinforcing the rationale behind the MRT's design.

3.3. Limitations and Conclusion

Our architecture, while effective, requires substantially more computational resources and time compared to alternatives, hindering its applicability in rapid scenarios like day-trading. Immediate reductions in iterations compromise model efficacy. More efficient diffusion architectures are essential for real-world use. In summary, our diffusion-based stock prediction architecture outperforms current models, presenting a promising avenue for improved stock market predictions and advancing research in this sector.

4. REFERENCES

[1] Ryo Akita and Kuniaki Y, "Deep learning for stock prediction using numerical and text info.," *15th ICIS*, 2016.

[2] K Chen and Zhou, "A LSTM-based method for stock returns," *2015 IEEE Big Data*.

[3] Xiao Ding and Junwen Duan, "Deep learning for event-driven stock prediction," *24th IJCAI*, 2015.

[4] Beatriz Vargas and A G Evsukoff, "Dl for stock market pred. from financial news," *2017 IEEE CIVEMSA*.

[5] R Sawhney and RR Shah, "Stock selection via sthan: A learning to rank approach," in *AAAI*, 2021.

[6] F Feng and Tat-Seng C, "Temp. relational ranking for stock pred.," *ACM Trans. on Info. Systems (TOIS)*, 2019.

[7] R Hyndman and Ralph D, *Forecasting with expo smoothing*, Springer Science & Business Media, 2008.

[8] G EP Box and G M Ljung, *Time series analysis*, John Wiley & Sons, 2015.

[9] K Bandara and S Smyl, "Forecasting across time series databases," *Expert systems with applications*, 2020.

[10] S Makridakis and V Assimakopoulos, "Stat. and ml forecasting methods," *PloS one*, 2018.

[11] K Rasul and R Vollgraf, "Autoregressive ddpm for multi. prob. time series forecasting," in *ICML*. PMLR, 2021.

[12] Y Tashiro and S Ermon, "Csdi for probabilistic time series imputation," *NIPS*, 2021.

[13] R Israel and T J M, "Can machines' learn'finance?," *Journal of Investment Management*, 2020.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Diffuion models," *NIPS*, 2020.

[15] D. Daiya and Che Lin, "Stock movement prediction and portfolio management via multimodal learning with transformer," in *ICASSP 2021*. IEEE, 2021.

[16] D. Daiya, M. Wu, and C. Lin, "Stock movement prediction that integrates het. data sources using dilated causal conv.," in *(ICASSP)*, 2020.

[17] Haomin Wen, Roger Zimmermann, and Yuxuan Liang, "Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models," *arXiv preprint arXiv:2301.13629*, 2023.

[18] I Assent and T Seidl, "Dtw for efficient similarity search in timeseries databases," *VLDB Endowment*, 2009.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] P Michel and G Neubig, "Are 16 heads really better than one?," *NIPS*, 2019.

[21] E Voita and I Titov, "Analyzing multi-head self-attention," *arXiv preprint arXiv:1905.09418*, 2019.

[22] J Lu and S Lee, "12-in-1," in *CVPR*, 2020.

[23] Yumo Xu and SB Cohen, "Stock movement prediction from tweets and historical prices," in *ACL*, 2018.

[24] Ramit S and RR Shah, "Spatiotemporal hcn for stock movement forecasting," in *ICDM*, 2020.

[25] Yingmei C and Xuanjing H, "Inc. corp. relationship via gcnn for stock price prediction," in *CIKM*, 2018.

[26] Raehyun K and Jaewoo K, "HATS: A hier. graph attention network for stock movement pred.," in *arXiv*, 2019.

[27] Robert Goodell Brown, *Smoothing, forecasting and prediction of discrete time series*, Courier Corporation, 2004.

[28] S Selvin and KP Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 (icacci)*. IEEE, 2017.

[29] V Pagolu and B Majhi, "Sentiment analysis for predicting stock market movements," in *2016 (SCOPES)*. IEEE.

[30] K Nguyen, THand Shirai, "Topic modeling based stock market pred.," in *ACL*, 2015.

[31] Z Hu and TY Liu, "Listening to chaotic whispers," WSDM '18.

[32] R Sawhney and RR Shah, "Deep attentive learning for stock mov. pred.," in *(EMNLP)*, 2020.

[33] F Feng and TS Chua, "Enhancing stock mov. pred. with adversarial training," *arXiv preprint arXiv:1810.09936*, 2018.

[34] S Carta and A Sanna, "Multi-dqn: An ensemble of deep q-learning agents for stock market forecasting," *Expert systems with applications*, 2021.

[35] Y Liu and C Liu, "Adaptive quant. trading: An imitative deep reinforcement learning approach," in *AAAI*, 2020.

[36] W Bao and Y Rao, "A dl framework for financial time series using lstm," *PloS one*, 2017.