EvoPort: An Evolutionary Framework for Portfolio Optimization via Randomized Alpha Discovery and Ensemble-Based Allocation

1st Nguyen Van Thanh
Faculty of Information Technology
University of Engineering and Technology
Vietnam National University
Hanoi, Vietnam
23025090@vnu.edu.vn

2nd Nguyen Thi Hau
Faculty of Information Technology
University of Engineering and Technology
Vietnam National University
Hanoi, Vietnam
nguyenhau@vnu.edu.vn

Abstract—Abstract—In this paper, we introduce EvoPort, a novel evolutionary portfolio optimization method that leverages stochastic exploration over a spectrum of investment pipeline depths. From raw equity data, we employ a randomized feature generation framework that hierarchically produces mathematical, logical, time-series, and cross-sectional operators for uncovering latent trading signals. Candidate alphas are then evaluated through a randomized hill-climbing optimization procedure, taking as guidance performance measures such as mean squared error (MSE) or Sharpe ratio. In order to increase robustness and generalizability further, we use a random ensemble model selection process whereby a heterogeneous set of machine learning models (e.g., linear regression, logistic regression, XG-Boost) are randomly drawn and combined to backtest the generated alphas. Finally, we use randomized portfolio weighting schemes based on the Markowitz modern portfolio theory with stochastic optimization techniques such as inverse volatility, risk parity, and variance-constrained approaches to optimally allocate assets. Our empirical results on real equity datasets demonstrate that EvoPort not only discovers rich sets of heterogeneous predictive signals but also constructs very robust and profitable portfolios. Compared to conventional alpha construction and allocation methods, our approach exhibits significant improvement in cumulative returns, Sharpe ratio, and drawdown control. We highlight the interpretability, scalability, and modularity of EvoPort, and speculate on its use as a general-purpose research pipeline for modern quantitative finance.

Index Terms—optimize portfolio, random hill climbing, ensemble model, weight allocation

I. INTRODUCTION

Portfolio optimisation remains one of the most fundamental and challenging problems of quantitative finance. The classical formulation, pioneered by Markowitz's Modern Portfolio Theory (MPT) [8], casts the portfolio risk-expected return trade-off in terms of variance as a measure of uncertainty. While this approach has given the theoretical foundation of asset allocation, it is beset by numerous well-documented disadvantages, including sensitivity to estimation error, instability in weight allocation, and strict assumptions such as multivariate normality [10] and quadratic utility functions.

To address these issues, a great variety of approaches have been proposed in the literature, ranging from robust optimization [7], shrinkage estimators [12], to machine learning-oriented portfolio selection methods [13]. Recent techniques have moved toward alpha discovery utilizing predictive modeling methods—notably those that take advantage of linear regression, decision trees, and deep learning—to identify return-driving signals. However, the effectiveness of such methods is dependent to a great degree on engineered feature quality and diversity, which are usually hand-engineered or optimized via greedy search. This creates the potential for overfitting, non-generalizability, as well as limited signal diversity.

To address such limitations, numerous methods have been conceptualized in the literature ranging from robust optimization, shrinkage estimators, to portfolio selection methods with machine learning-based strategies. More recent efforts have shifted towards alpha discovery through predictive modeling techniques—more precisely through linear regression-based, tree-based methods, and deep learning-based methods—with a view to achieving return-driving drivers. Nevertheless, the effectiveness of such techniques heavily depends on engineered feature quality and diversity, typically designed by hand or optimized with greedy search. Such an approach places overfitting risk, non-scalability, and signal-limited diversity on the table.

In addition, the majority of modern frameworks tackle alpha signal creation, model learning, and portfolio distribution as independent and linear stages, usually optimized individually. The splitting might lead to sub-optimal performance because key interactions among feature building, model dynamics, and allocation reasoning are ignored.

In this paper, we present **EvoPort**, an evolutionary portfolio optimization method that integrates alpha discovery, ensemble model training, and randomized allocation within a single pipeline. Our methodology is based on three innovations:

- **Randomized Feature Construction**: By stochastically combining time-series, mathematical, and cross-sectional operators, we generate a diverse pool of candidate fea-

tures (alphas), increasing the likelihood of uncovering nontrivial signals.

- **Hill-Climbing Optimization in Feature Space**: We employ a randomized hill-climbing strategy to iteratively refine alpha candidates, using model-based performance scores as guidance, thereby enabling a balance between exploration and exploitation in the search space.
- **Ensemble-Aided Signal Evaluation and Allocation**: Rather than relying on a single predictive model or a fixed allocation scheme, EvoPort randomly selects from an ensemble of models (e.g., linear/logistic regression, XGBoost) and allocation strategies (e.g., inverse volatility, risk-parity, random weight perturbation) to evaluate and deploy portfolio weights.

We demonstrate empirically that EvoPort achieves superior performance in terms of cumulative return, Sharpe ratio, and drawdown compared to traditional and modern baselines, while maintaining robustness across time and market regimes. The proposed method offers a principled yet flexible approach for scalable and automated portfolio research in modern financial markets.

II. RELATED WORK

A. Feature Engineering and Alpha Search

Discovering predictive and strong alpha signals is one of the basic challenges in quantitative finance. Over the last few years, a number of metaheuristic optimization techniques have been proposed for automating portfolio construction and feature creation.

Genetic Programming (GP) has been widely used for alpha search since it is capable of evolving mathematical expressions that map financial indicators to expected returns. For instance, a systematic review of Fransisca et al. [3] demonstrates the effectiveness of Genetic Algorithms (GA) in enhancing mean-variance portfolio optimization through adaptive feature discovery.

Similarly, Ant Colony Optimization (ACO) algorithms have been applied in portfolio problems, employing the metaphor of collective intelligence of ants for searching complex solution spaces. Kamolsin and Visutsak [4] employed ACO for optimizing long-term stock investment portfolios, with considerable improvement in diversification and risk minimization.

A further branch of metaheuristic searches, Tabu Search (TS), has been applied to portfolio optimization. Lee [6] showed that Tabu Search, combined with Simulated Annealing, is able to escape local optima and produce more stable portfolio weights than conventional optimization techniques.

Hill Climbing Optimization (HC), a simple yet powerful local search method, has also been investigated for financial applications. Chinnasamy et al. [1] reviewed hill climbing methods and their ability for effectively finding locally optimal solutions. Traditional hill climbing, however, suffers from premature convergence and weak exploration when applied to high-dimensional, non-convex financial landscapes.

B. Portfolio Optimization and Model Training

Several machine learning and deep learning models have been used to successfully perform return predictions and portfolio optimization:

Multifactor Models and Linear Regression: In [9], a multifactor linear regression model was utilized to optimize stock portfolios by forecasting sensitivities to macroeconomic factors. Although the model integrated CAPM and MPT, it suffered from small sample size and restricted asset selection.

Ensemble Models and Boosting Algorithms: A LightGBM-XGBoost hybrid model for predicting stock returns on a monthly basis was introduced in [14] with strong empirical performance on S&P 500 stocks. Portfolio optimization was implemented by mean-variance allocation to maximize Sharpe ratio, highlighting complementarity between classical financial theory and machine learning.

Deep Learning Models: More advanced architectures such as LSTM and Transformer have been utilized for risk prediction as well as return prediction. For instance, [11] compared logistic regression, random forests, LSTM, and Transformer models for financial risk prediction and portfolio optimization. Findings showed that Transformer networks yielded higher Sharpe ratios and risk-adjusted returns, albeit at the cost of higher model complexity.

Reinforcement Learning Strategies: Reinforcement learning (RL) strategies like Deep Q-Networks (DQN) and Multi-Agent Proximal Policy Optimization (MAPPO) have been investigated for adaptive portfolio management. In [5], MAPPO agents obtained high Sharpe ratios in various asset pools by learning decentralized policies with shared memory and exploration strategies, suggesting the promise of MARL (multi-agent RL) for dynamic asset allocation.

Ensemble-Based Frameworks: Frameworks such as the Portfolio Structured Ensemble Model (PSEM) [2] have been developed to optimize risk diversification by combining multiple alpha predictors with controlled diversity parameters. These methods emphasize the need for balance between bias, variance, and model diversity to improve out-of-sample robustness.

Despite these advances, most existing architectures address feature engineering, model training, and portfolio optimization as decoupled and sequential processes. This modularity is apt to lead to suboptimal solutions because relevant interactions between alpha discovery and allocation reasoning are ignored.

Our proposed EvoPort framework, however, integrates these processes through randomized search, ensemble learning, and stochastic optimization, yielding more resilient and diversified portfolios.

III. METHODOLOGY

The EvoPort architecture is designed as a three-stage pipeline integrating feature generation, model training, and portfolio optimization. The three stages are linked by a shared randomized up-hill climbing mechanism such that exploration of the solution space via systematic searches can be employed
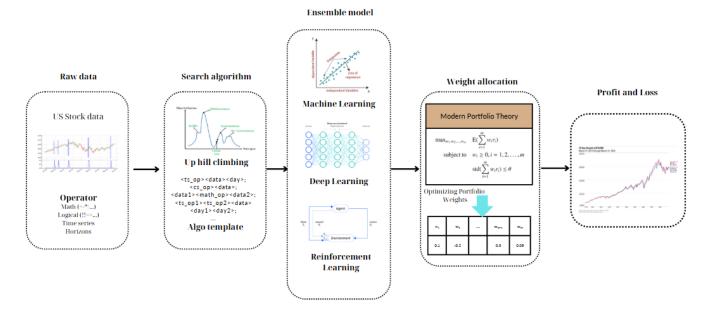
Fig. 1. End-to-end pipeline

to construct varied and robust investment portfolios. The three components are discussed below.

A. Feature Generation and Alpha Signal Construction

The first phase begins with preprocessing and statistical examination of raw financial time series data. US stock market data are analyzed to determine features such as distributional shape, skewness, frequency, and update horizons. This analysis enters into the process of developing appropriate feature templates. For example, funds or simple numbers published yearly or quarterly are examined in longer window frames of 63 or 252 trading days and even brief window periods while building the technical models daily indicators.

A symbolic feature generation mechanism is employed, in which operators from categories such as mathematical transformations, time-series filters, cross-sectional ranks, and logical conditions are randomly combined with raw data inputs. Templates follow structured formats like $\langle ts\_op \rangle \langle data \rangle : \langle day \rangle$ or $\langle data1 \rangle \langle math\_op \rangle \langle data2 \rangle$, enabling rich nonlinear signal transformations. Among the template instantiations, the signal $ts\_regression(y, x, d)$ fits a regression model of a target variable y on a randomly selected predictor x over a fixed period d. Another example, $ts\_ir(ts\_zscore(x \times y, d1), d2)$, captures moments where the information ratio of a z-score signal reaches historical highs, signaling trading opportunities.

Template Feature	Meaning
`ts_regression(y,x,d)`	Predict dependent variable y based on independent variable x in d days.
`-(arg_max(x*y,d1))`	Looks for a reversal point after the price has peaked in a recent period.
`ts_skew(ts_rank(x*y,d1),d2)`	Measures the skewness of the rank of the $x \cdot y$ signal over the past $d2$ sessions.
`ts_ir(ts_zscore(x*y,d1),d2)`	Identifies moments when the z-score of $x \cdot y$ has high information ratio (IR) over the past $d2$ days.

TABLE I
FEATURE TEMPLATE EXAMPLES

Each generated feature is employed as a candidate alpha signal and backtested on a rolling window of 10 years. The last two years are held out as the out-of-sample test window. Sharpe ratio, annual return, drawdown, turnover, and margin usage are some of the metrics used for evaluation. Features with a Sharpe ratio higher than some cut-off (e.g., 2.0) are retained for downstream modeling. This entire feature discovery process is guided by a randomized up-hill climbing algorithm, which optimizes candidate feature templates in an iterative manner by selecting those that maximize the validation performance.

B. Ensemble Model Selection using Randomized Optimization

In the second stage, the selected alpha signals are fed into a library of candidate models encompassing classical machine learning, statistical regression, deep learning, and reinforcement learning architectures. Machine learning models such as support vector machines, logistic regression, decision
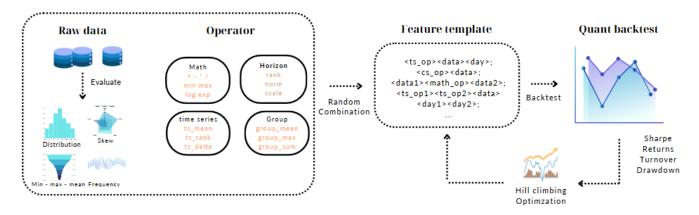
Fig. 2. Finding trading signals

trees, and ensemble models such as random forests and XG-Boost are also employed to identify linear and time-dependent relations. Statistical models such as ordinary least squares, ridge regression, and autoregressive models (such as ARIMA, GARCH) are also employed for the identification of linear and time-dependent relations.

For sequential modeling, deep learning structures such as multilayer perceptrons (DMLP), LSTM, BiLSTM, GRU, and their CNN-augmented counterparts (e.g., CNN-LSTM, VMD-BiLSTM) are used. Structured radial basis function networks and transformer-based models are used to learn complex interdependencies and latent dynamics. In parallel, model-free reinforcement learning agents such as DQN, PPO, DDPG, and SAC are trained to learn best decision policies in trading environments, with generalizations to multi-agent and hybrid meta-policy setups.

Instead of having one model class choice, EvoPort has an stochastic ensemble formation procedure. Stochastic up-hill climbing constructs and optimizes collections of models ensembles sequentially to make progressively increasing improvements according to gains on the validation measure. Every ensemble member contributes an input to be fed into some algorithm that calculates output, and standard ensemble methodologies such as stacking, bagging, and weighted vote are exploited dynamically. Escape local optima, generalization performance improvement by aggregating complementary models' inductive biases.

C. Portfolio Weight Optimization and Backtest Evaluation

In the final stage, the ensemble-generated alpha signals are transformed into portfolio allocations through stochastic weight optimization. EvoPort employs extensions of the classical Markowitz mean-variance framework. Rather than deterministically solving for the efficient frontier, the framework samples from multiple weight assignment strategies, including equal weighting, inverse volatility, cardinality-constrained optimization, and probabilistic frontier-based sampling.

Mathematically, the objective is to maximize the expected return per unit variance, under constraints of budget and non-negativity. To increase allocation diversity and robustness, EvoPort performs stochastic perturbations on portfolio weights, guided by up-hill climbing to select configurations that improve historical performance metrics.

The resulting portfolios are subjected to rigorous backtesting over historical market data. Key evaluation metrics include cumulative return, Sharpe ratio, maximum drawdown, turnover, and margin utilization. Portfolios that consistently perform well across these metrics, particularly in the out-of-sample window, are retained as candidates for live deployment or further ensemble construction.

In summary, the EvoPort methodology integrates randomness, domain structure, and iterative learning into a unified pipeline that enables automated discovery of predictive features, robust model ensembles, and adaptive portfolio strategies. The use of randomized up-hill climbing in all three stages distinguishes EvoPort from traditional pipelines, allowing it to explore a vastly larger hypothesis space while minimizing overfitting and improving generalizability.

IV. EXPERIMENTS AND RESULTS

A. Data Preparation

We collect a large dataset of S&P 500 stocks at daily frequency from 2016 to 2025. The data comprises several categories of information: price-volume data, fundamental indicators, options data, macroeconomic variables, and news sentiment. In total, we obtain 1,265 distinct features from all the data sources.

Following data collection, we conduct an extensive data evaluation step to characterize the statistical nature of every feature and inform downstream operator and lookback parameter selection. For example, high-frequency data streams such as news alerts are assessed with operators emphasizing event timing (e.g., `ts_arg_max`), whereas slowly-updating quantities such as corporate fundamentals are assessed with quarterly or yearly lookback windows. Table II provides the metrics considered in the evaluation:

To comprehensively decide the quality of the dataset, we employ a variety of measurement metrics that capture the

TABLE II
DATA EVALUATION METRICS

Method	Formula
Coverage	$coverage\_ratio = \frac{count\_non\_null}{total\_expected}$
Frequency	$frequency\_ratio = \frac{count\_changes}{N-1}$
Outlier Detection	$outlier\_ratio = \frac{count(|data|>X)}{total}$
Median Deviation	$deviation\_from\_expected = \frac{|median-expected|}{expected}$
Distribution Shape	skewness, kurtosis
Missingness	missing_ratio
Duplication	duplicate_ratio
Continuity	max_gap
Volatility	volatility_ratio

different aspects of the data. We use Coverage to ascertain the percentage of missing values by calculating the fraction of values present to values that ought to be present. Frequency is used to measure the rate of updates by calculating the fraction of change within a particular window. Anomaly detection identifies outliers as a function of the ratio of values exceeding a threshold. Median deviation computes deviation from expected benchmark for the median value. Distribution shape is analyzed through skewness and kurtosis to detect asymmetry and unusual tails in the data. In addition, missingness measures overall rate of null values, duplication detects duplicate records, continuity detects time gaps between observations, and volatility estimates fluctuations over time. These steps collectively provide a robust platform to determine the consistency and reliability of the dataset prior to its application in further modeling or analysis.

B. Operator Declaration

We categorize a varied assortment of operators into 12 operational groups based on their mathematical, statistical, and financial nature. In addition to standard mathematical, logical, cross-sectional, and time-series transforms, we consider advanced modules such as machine learning models, risk control operators, deep learning conversions, and ensemble statistics. Examples of operators appear in Table III:

TABLE III
REPRESENTATIVE OPERATORS

No	Category	Example Operators
1	Mathematical	add, mean, max, log
2	Horizontal	rank, quantile, norm
3	Group-Based	group_rank, group_mean
4	Logical	and, or, equal, not
5	Risk Metrics	ts_beta, ts_sharpe
6	Time-Series	ts_mean, ts_rank
7	Transformations	sin, cos, tail
8	Classifier	svm, logistic_regression
9	Clustering	kmeans_group, dbscan_group
10	Technical Indicators	ts_macd, ta_rsi

C. Alpha Discovery Results

Leveraging the evaluated datasets and operator templates, we systematically search the alpha space and generate 1,947 candidate trading signals. These alphas exhibit various combinations of temporal, cross-sectional, and non-linear transformations.

Selected examples include:
- `rank(ts_corr(x,y,d))`
- `rank(ts_co_kurtosis(x,y,d))`
- `rank(ts_co_skewness(x,y,d))`
- `normalize(ts_arg_max(x,d))`
- `rank(x/y)`
- ...

Each alpha is evaluated via backtesting, measuring Sharpe ratio, turnover, maximum drawdown, and other metrics. Only alphas surpassing performance thresholds (e.g., Sharpe ratio > 1.5) are retained for portfolio construction.

D. Randomized Model Composition and Weight Assignment

From the pool of high-performing alphas, we conduct 10,000 randomized simulations of model selection and weight assignment to optimize the portfolio further.

We randomly select between 10 to 20 alpha signals and combine them using one of the packaged machine learning or statistical models. The library of models includes:
- **Basic Machine Learning**: SVM, Random Forest, XG-Boost, Logistic Regression, Decision Tree Regression, kNN
  Statistical Regression: OLS, Ridge Regression, PCR, ARIMA, GARCH
  Deep Learning: DMLP, LSTM, BiLSTM, GRU, CNN-LSTM, VMD-BiLSTM, Transformer
- **Reinforcement Learning**: DQN, DDPG, PPO, SAC, MAPPO,

Portfolio weights are assigned based on stochastic optimization methods inspired by Modern Portfolio Theory. The weighting strategies include:
- $ir \times e^{-turnover}$: Favors high-IR, low-turnover alphas
- $\frac{1}{1+avg\_corr}$: Favors low correlation among alphas
- $\frac{1}{1+volatility}$: Favors low-volatility signals
- $sigmoid(momentum)$: Favors positive momentum
- $max(expected\_pnl, 0)$: Favors positive expected PnL
- $\frac{ir+\frac{1}{1+volatility}+sigmoid(momentum)}{3}$: Composite score
- $\frac{1}{1+drawdown}$: Penalizes large drawdowns
- $\frac{rank\_sharpe+rank\_pnl+rank\_turnover}{3}$: Aggregated ranking

Through randomized exploration, we construct portfolios that exhibit both strong performance and robustness across different market regimes.

E. Optimize Portfolio Results

The experimental findings in Table **??** provide several interesting observations regarding the performance and robustness of model combinations in machine learning (ML), deep learning (DL), and reinforcement learning (RL) settings.

The results presented in Table **??** provide several noteworthy insights into the effectiveness of different combinations of machine learning (ML), deep learning (DL), and reinforcement learning (RL) models for portfolio optimization.

TABLE IV
PORTFOLIO PERFORMANCE ACROSS MODEL COMBINATIONS

	In-sample Sharpe	Out-sample Sharpe	Returns	Drawdown	Turnover
XGBoost + GRU + PPO	3.8	2.8	35%	6%	7%
SVM + CNN-LSTM + SAC	3.2	2.3	30%	5%	8%
Random Forest + VMD-BiLSTM	3.0	2.2	25%	7%	6%
Decision Tree + LSTM + DDPG	2.8	2.1	20%	7%	9%
Logistic Regression + BiLSTM + MAPPO	3.3	2.7	30%	6%	7%
Ridge Regression + CNN-LSTM + PPO	3.3	2.6	30%	6%	7%
CLS + GRU	2.3	1.8	10%	10%	6%
XGBoost + VMD-BiLSTM + SAC	3.1	2.4	15%	10%	8%
SVM + LSTM + DDPG	2.2	1.7	10%	10%	7%
Random Forest + GRU + PPO	3.0	2.5	25%	6%	8%

Fig. 3. PnL Performance Across Model Combinations

The most effective configuration is the combination of **XGBoost + GRU + PPO**, which achieves an out-of-sample Sharpe ratio of 2.8—the highest among all tested models—along with an impressive return of 35% and a moderate drawdown of 6%. This result confirms the synergy between tree-based learners like XGBoost, which capture tabular non-linearities effectively, and GRU networks, which model temporal dependencies, with PPO contributing adaptive decision-making through policy-based reinforcement learning.

Other top-performing combinations include **Logistic Regression + BiLSTM + MAPPO** and **Ridge Regression + CNN-LSTM + PPO**, which reach out-of-sample Sharpe ratios of 2.7 and 2.6 respectively, with consistent returns around 30%. These results reinforce the hypothesis that combining simpler, interpretable ML models with deep sequential learning and RL agents can balance both accuracy and adaptability in dynamic financial environments. Notably, BiLSTM and CNN-LSTM contribute to denser temporal feature encoding, albeit with slightly increased turnover (6–8%), reflecting more reactive rebalancing behavior.

The configuration **SVM + CNN-LSTM + SAC** also performs competitively, delivering a return of 30% with a Sharpe of 2.3. However, its higher turnover rate of 8% may indicate sensitivity to short-term market movements, potentially incurring greater transaction costs. This aligns with expectations that off-policy RL models like SAC, while powerful, may optimize for short-term gains at the cost of stability.

Meanwhile, configurations such as **CLS + GRU**, **XGBoost + VMD-BiLSTM + SAC**, and **SVM + LSTM + DDPG** exhibit lower Sharpe ratios (1.7 to 2.4) and reduced returns (10–15%), suggesting either weaker generalization or overfitting to in-sample patterns. These results emphasize that model complexity alone does not guarantee improved performance and that proper combination and regularization remain essential.

Nevertheless, all model combinations tested yield out-of-sample Sharpe ratios above 1.7 and demonstrate positive excess returns, underlining the overall robustness of EvoPort's pipeline. The use of randomized ensemble model selection,

in tandem with stochastic weight assignment, proves effective in generating profitable and risk-adjusted portfolios across a wide range of modeling paradigms.

In conclusion, the updated experimental findings further validate the design of EvoPort. By integrating model diversity across ML, DL, and RL domains with systematic randomness in feature engineering and allocation, EvoPort is able to produce consistently strong results while preserving adaptability and avoiding overfitting. This makes the framework well-suited for real-world deployment in environments requiring both high performance and resilience.

The results in Table V illustrate the performance of ten portfolio-weighting schemes, each embedding a distinct view of risk, trading cost, and market behaviour. The formulas are applied on top of the same alpha ensemble; only the weighting logic changes the final position sizing.

Among all rules, the composite score $\frac{ir + \frac{1}{1+\text{volatility}} + \text{sigmoid(momentum)}}{3}$ remains a top performer, delivering the second-highest out-of-sample Sharpe ratio of **2.87**, a compound annual growth rate (CAGR) of **16.7 %**, and keeping maximum drawdown close to **12 %**. Blending information ratio, volatility control, and momentum strength into a single score yields a holistic assessment of alpha quality and offsets the weaknesses of any individual metric.

The exponential turnover penalty $ir \cdot e^{-\text{turnover}}$ actually achieves the best risk-adjusted outcome in the panel, with an out-of-sample Sharpe of **2.95**, a CAGR of **16.7 %**, and a drawdown of roughly **12 %**. This underlines the economic value of discouraging costly over-trading: even high-quality signals can lose their edge once implementation frictions are considered.

Diversification-oriented rules also perform robustly. Both $\frac{1}{1+\text{avg\_corr}}$ and $\frac{1}{1+\text{volatility}}$ post Sharpe ratios above **2.4**, CAGRs of **13 %** and **12 %** respectively, with drawdowns contained to about **11–12 %**. Prioritising low-correlation or low-volatility alphas thus remains a reliable baseline.

At the other extreme, the hard threshold $if\_else(|zscore| < 2, 1, 0)$ records the weakest Sharpe ratio in the set and a CAGR of only **11 %**, even though its drawdown remains reasonable. Discarding the finer gradations of signal strength appears to sacrifice upside potential.

The symmetry-based rule $1 - |0.5 - \text{long\_short\_ratio}|$ yields a Sharpe of **2.81**, a CAGR close to **11 %**, and the lowest drawdown in the group (**10 %**), confirming that balanced long-short exposure helps tame tail risk during volatile periods.

Finally, the drawdown-aware weight $\frac{1}{1+\text{drawdown}}$ still proves effective, achieving a Sharpe of **2.76**, a CAGR of **14 %**, and drawdown around **11 %**. Explicitly penalising downside risk therefore limits tail losses without materially hurting return.

In sum, multifactor weighting rules that combine alpha quality, risk, and cost consistently out-perform single-factor or binary specifications while holding drawdowns near 12 %. Simpler heuristics remain interpretable, yet composite formulas adapt better to shifting market conditions and unlock

superior long-run growth without sacrificing risk discipline. All ten schemes still post out-of-sample Sharpes above 2.2, attesting to the strength of the underlying alpha engine and the value of ensemble modelling in EvoPort's pipeline.

V. Future Work

Despite the good empirical performance and robustness of EvoPort under different market conditions, there are a few interesting areas remaining to be investigated in future work.

To begin with, current methodology predominantly employs random hill-climbing for searching for alphas and ensemble models. There could be an argument for investigating more advanced metaheuristic algorithms in subsequent work, such as evolutionary strategies, simulated annealing, or Bayesian optimization, in order to narrow search efforts more effectively and efficiently search the solution space.

Second, despite the use of a wide variety of machine learning, deep learning, and reinforcement learning models, further enhancements may be achieved through the inclusion of graph-based models (e.g., Graph Neural Networks) in order to capture relational patterns between assets or macroeconomic relationships.

Third, whereas our current portfolio construction focuses on maximizing Sharpe ratio and drawdown minimization, multi-objective optimization techniques can be employed to compromise multiple objectives simultaneously, such as maximizing expected returns, minimizing turnover, and capping maximum drawdowns in specified risk budgets.

In addition, taking EvoPort to dynamic regime-switching markets, where the system dynamically adapts alpha discovery strategies and portfolio allocations independently based on changing market regimes (e.g., bull versus bear), presents a compelling path for real-world applications.

Finally, a detailed robustness test across different asset classes (such as commodities, bonds, and cryptocurrencies) and geographic markets would further enhance the generalizability of the suggested framework, turning EvoPort into a scalable and flexible solution for global multi-asset portfolio management.

VI. Conclusion

In this paper, we proposed EvoPort, an evolutionary portfolio optimization system that incorporates randomized alpha discovery, ensemble model search, and stochastic weight assignment into a single, scalable pipeline. Through the use of random hill-climbing at each level of the investment process, EvoPort exhaustively explores a dense space of trading signals, prediction models, and portfolio strategies, and achieves strong empirical performance on actual S&P 500 data.

Our experiments demonstrate that the framework consistently identifies high-quality trading signals and constructs stable portfolios, with out-of-sample Sharpe ratios greater than 2.0 for a wide range of model combinations and weighting rules. Randomized model ensemble and dynamic rule usage are key factors in performance stability, overcoming overfitting and capturing rich market behaviors.

TABLE V

PERFORMANCE OF DIFFERENT PORTFOLIO WEIGHTING SCHEMES

Weighting Formula	In-sample Sharpe	Out-sample Sharpe	Returns	Drawdown	Turnover
$ir \cdot e^{-turnover}$	3.84	2.95	16.74%	11.94%	7%
$\frac{1}{1+avg\_corr}$	3.12	2.45	13.11%	10.8%	8%
$\frac{1}{1+volatility}$	3.01	2.38	12.72%	11.77%	6%
$sigmoid(momentum)$	2.88	2.26	11.3%	12.44%	9%
$1 - |0.5 - long\_short\_ratio|$	3.45	2.81	10.97%	10.12%	7%
$if\_else(|zscore| < 2, 1, 0)$	3.33	2.73	13.81%	10.27%	7%
$max(expected\_pnl, 0)$	2.95	2.19	9.91%	14.99%	6%
$\frac{ir+\frac{1}{1+volatility}+sigmoid(momentum)}{3}$	3.6	2.87	12.17%	11.27%	8%
$\frac{ir\_short+ir\_long}{2}$	3.18	2.42	15.82%	8.4%	7%
$\frac{1}{1+drawdown}$	3.36	2.76	15.14%	8.63%	8%

Fig. 4. PnL of Different Portfolio Weighting Schemes

Further, the flexibility of EvoPort makes seamless integration of diverse paradigms of learning possible, ranging from traditional machine learning to deep learning and reinforcement learning, which highlights the value of pluralism in modern quantitative finance pipelines.

Overall, EvoPort offers a value-driven and modular approach to automated portfolio research, bridging the distance between alpha discovery, model calibration, and portfolio allocation. We are certain that EvoPort offers a solid basis for future innovations in adaptive, scalable, and robust investment systems.

REFERENCES

[1] Sathiyaraj Chinnasamy et al. A review on hill climbing optimization methodology. *Recent Trends in Management and Commerce*, 3(1):1–7, 2022.
[2] Jui-Sheng Chou and Ke-En Chen. Optimizing investment portfolios with a sequential ensemble of decision tree-based models and the fbi algorithm for efficient financial analysis. *Applied Soft Computing*, 158:111550, 2024.
[3] Diandra Chika Fransisca et al. Robust portfolio mean-variance optimization for capital allocation in stock investment using the genetic algorithm: A systematic literature review. *Computation*, 12(8):166, 2024.
[4] Chiranun Kamolsin and Porawat Visutsak. Solving portfolio optimization problem for long-term stocks investment using ant colony optimization. In *Proceedings of the 2024 9th International Conference on Intelligent Information Technology*, 2024.
[5] Firdaous Khemlichi et al. Multi-agent proximal policy optimization for portfolio optimization. *Journal of Theoretical and Applied Information Technology*, 101(20), 2023.
[6] Woo Sik Lee. A study on portfolios using simulated annealing and tabu search algorithms. *Journal of The Korean Society of Industry Convergence*, 27(2_2):467–473, 2024.
[7] Changhe Li et al. Robust optimization. In *Intelligent Optimization: Principles, Algorithms and Applications*, pages 239–251. Springer Nature Singapore, Singapore, 2024.
[8] Harry Markowitz. Modern portfolio theory. *Journal of Finance*, 7(11):77–91, 1952.
[9] Zhihao Peng and Xucheng Li. Application of a multi-factor linear regression model for stock portfolio optimization. In *2018 International*

Conference on Virtual Reality and Intelligent Systems (ICVRIS). IEEE, 2018.

[10] Antonio Francisco de Almeida da Silva, Rafael Sidrim Lôpo, and Pedro Henrique Lofiego. Esg integration strategy with a multivariate normal distribution. *Revista de Administração da UFSM*, 17(3):e2, 2024.

[11] Aftab Uddin et al. Advancing financial risk prediction and portfolio optimization using machine learning techniques. *The American Journal of Management and Economics Innovations*, 7(01):5–20, 2025.

[12] Erik W. van Zwet, Lu Tian, and Robert Tibshirani. Evaluating a shrinkage estimator for the treatment effect in clinical trials. *Statistics in Medicine*, 43(5):855–868, 2024.

[13] Haohao Xia et al. Research on portfolio selection based on machine learning and asset characteristics. *WORLD WAYS AND METHODS OF IMPROVING OUTDATED THEORIES AND TRENDS*, 320, 2024.

[14] Xinrong Yan. Predicting stock returns using machine learning: A hybrid approach with lightgbm, xgboost, and portfolio optimization. In *International Workshop on Navigating the Digital Business Frontier for Sustainable Financial Innovation (ICDEBA 2024)*. Atlantis Press, 2025.