Scientific Opinion on the Invesdwin Platform

Hans-Wolfgang Loidl

Heriot-Watt University, Edinburgh, UK

***Abstract*—This document presents a scientific opinion on the FinTech platform "Invesdwin." It considers the academic foundations of the platform, its expressive power, and its performance, in the context of the wider FinTech domain. The document reflects the status of the platform as of December 2024.**

I. Introduction

The Invesdwin platform is a FinTech platform to explore a wide range of different portfolio trading strategies, and to assess the market performance of these strategies on historical and live streaming data. It features a high-performance compute engine, a flexible machine-learning engine, and low-latency implementation of data-streams. In the finest granularity the market data is captured on a tick-basis. The platform has been used on a range of financial instruments such as stocks, market indices, foreign exchange rates, and futures.

An outstanding feature of the platform is its computational performance and this is discussed in Section V. As a bottom line of detailed performance comparisons, Invesdwin is 6.5x times faster than the fastest contender platform that is currently available to research (BuildAlpha). As another novelty, it defines an "expression language" for market interactions, that features building blocks for trading strategies and portfolio management to achieve a modular design. Example expressions for entering trades are "enter at a certain price", or, as part of a condition for taking action, "examine the moving average over 200 days". The expressive power of this rule-based expression language is summarised in Section VI.

II. Context

The current explosion of AI research opens up avenues for exploiting sophisticated machine-learning techniques in the domain of FinTech. The most interesting such technique is the automatic generation of trading strategies, that yield a benefit across a wide range of different market scenarios. Key challenges for such a platform are: (1) the efficient handling of large data volumes; (2) the automatic generation of candidate strategies; and (3) the efficient evaluation of a large number of trading strategies. Addressing these challenges requires a high-performance compute engine, with optimisations across the software stack, as well as a flexible language for expressing trading strategies and higher-level decisions. Delivering on both of these key aspects is one main strength of the platform under discussion: the Invesdwin platform.

III. Foundations

The machine-learning (ML) engine of the Invesdwin platform builds on e.g. Differential Evolution [1]. This ML technique quickly converges to local optima that can be evaluated as generated strategy candidates for selecting diversified portfilios. In this FinTech context it is desirable to generate a high number of strategy candidates and to evaluate all of them in a high-performance compute engine. Flexibility is one of the major advantages of this ML technique. Specifically, it can be applied to data distributions that are not differentiable, i.e. where it is in general not possible to determine a gradient of ascent or descent for a given point in the data distribution. This property makes it suitable for a wide range of application areas, with FinTech being a prominent and timely one. The downside of dealing with non-differentiable distributions is the lack of guidance in performing a search, provided by the distribution per-se. To counter this conceptual limitation, a range of refinements and alternatives have been developed in literature and the Invesdwin platform implements those most effective for the FinTech domain.

The core compute engine for strategy evaluation is implemented in Java. It leverages open-source libraries where possible. In many cases, custom developed components implement performance-critical parts of the platform, sometimes as replacements for less performant existing libraries. This attention to detail when it comes to performance issues is key to delivering a high-performance platform. Furthermore, integration of reusable libraries facilitates extensive features for report generation and visualisation to validate and monitor extensive automations. This combination ensures portability

of the core engine, while guaranteeing high-performance execution of critical components and also keeping high developer productivity. Bindings of the core engine to other languages, most notably Python, R, Matlab, Julia and Haskell, allow the encoding of specific functionality in strategy evaluation or generation in external languages. This adds to the flexibility and interoperability of the platform and allows to leverage open-source machine learning solutions beyond the Java ecosystem.

IV. Market Performance of Invesdwin

The Invesdwin platform has been designed as a powerful tool for portfolio managers to make decisions on portfolio composition based on core market data. It provides two core components of use for managers: (1) a powerful compute engine for running strategies on sizable market data; and (2) a flexible, automated generator of trading strategies. Each of these is supported with a user interface as well as an API to facilitate automation. The compute engine in (1) plays a crucial role in running a given strategy, either composed via expressions by the portfolio manager, or automatically generated. Developers can build extensions for the expression language or take full control with the underlying fast, event-based strategy API. Being a high-performance implementation, strategies can be executed quickly and classified through the powerful reporting back-end mechanism. The automated generation in (2) is a crucial enabler for fully automated machine-learning techniques, closing the loop from strategy generation, evaluation, and classification of results. While top-performing strategies are a natural focus in evaluating candidate strategies, a wider impact is delivered by strategies that are robust. Thus, using the engine in robustness analysis is a key direction of the current research performed with this platform. In particular this research focuses on hybrid ML techniques that e.g. combine Differential Evolution, to generate strategy candidates, with domain-specific ML techniques such as SFA or DEA for selecting portfolios from these candidates. Such robustness analyses focus on reducing the variability of the market performance, while still delivering competitive performance results. Addressing this challenge with empirical results is a key step in the acceptance of automated techniques for portfolio traders. Linking decisions made by the strategies with human-understandable explanation is another key step in this process and supported by the logging and monitoring component of the platform. This is an instance of the wider research challenge of explainable AI, in this instance in the domain of FinTech, communicating to specialised portfolio managers.

V. Computational Performance of Invesdwin

An outstanding feature of the Invesdwin platform is its computational performance. In scientific measurements comparing the performance of decision making, generating and evaluating strategies based on mathematical rules, the platform outperformed one of the best strategy generators (GeneticSystemBuilder) by a factor of 596x. When optimising classical trading strategies, the platform outperformed one of the fastest contenders (Zorro) by a factor of 148x. These performance results are discussed in detail in [2].

These impressive numbers are due to thorough performance tuning and careful design. Outstanding features that contribute to the performance gains achieved by the platform are in-memory management of data (in particular, efficient caching), usage of a custom NoSQL database, on-the-fly data compression, and moderate use of multi-threading. Details of these techniques can be found in the joint journal paper [2].

The investigated usage profile for the platform is high-performance strategy evaluation, typically in batch jobs. This setup focuses on processing as many evaluations in a given time span as possible, and notably is different to low-latency trading where the focus is on minimising the response time to one individual trading decision. This focus is motivated by the highlighted use case of the platform for portfolio managers. However, the speedup is partially achieved by applying optimisation techniques from low-latency trading to this use case of strategy simulations via backtesting.

VI. An Expression Language Embedded in Invesdwin

A core aspect of the flexibility provided by the Invesdwin platform is its expression language: a domain-specific language describing conditions for and actions taken in interactions with the underlying platform [3]. While general platforms all need to support some form of describing decisions on which to e.g. enter a trade, most of these platforms limit the scope of conditions that can be described. In the Invesdwin platform, a powerful, underlying language is defined as the basis for these decisions, with well-defined semantics. Through a Java-based user interface, this “expression language” is accessible to and programmable by the expert user. Through a clear API it can be accessed programmatically, and this facility is used in the ML component of the platform for strategy generation.

The well-defined nature of the expression language also opens up the opportunity for optimisations in evaluating the conditions for actions taken. Fundamentally, the expression language is a rule-based language, with guards triggering the execution of an action, typically

buy/sell. The components of the guards draw on a rich set of building blocks, examining basic market parameters and performing functions over these parameters. The efficient evaluation of these guards is drawing on caching and other memory-based techniques, built into the core Java-based compute engine. The modular nature of these building blocks facilitates the automatic generation of rules, which in turn enables machine learning approaches for trading strategies.

Examples for the flexibility of the expression language are the possibility to define expressions for building blocks of risk management such as position sizing, weighting of portfolio components, lifecycle management with equity curve trading, and reacting to market changes through regime detection. Diversification can be automated via expressions for portfolio selection of instrument universes, time frames, or strategy candidates. Functions can be prefixed to select data streams on other instruments, time frames, fundamentals, or alternative data. Data streams can aggregate bars via custom expressions to define at which granularity data will be processed. This can enable an approach of pattern detection within the data-stream (e.g. for detecting extrema). The possibility of pre-calculation and caching of the data-streams further improves computational performance for custom strategies. The user/developer can define his/her own complex functions, which can perform calls to external machine learning techniques. This enables the exploration of advanced, hybrid ML techniques for improving market performance. Any building block can potentially be generated similar to entry/exit rules. This allows to not only generate trading strategies, but also risk management or any other fund management principle. This facilitates meta optimisation, vertical and horizontal integration, as well as the generation of hybrid ML approaches spanning multiple building blocks. In summary, these features of the expression language enable research on more and higher level decisions of fund managers than previously possible.

VII. Performance Trader

The Performance Trader is a proprietary extension of the Invesdwin platform, owned and operated by 123 Invest Group. It focuses on real-time applications for quantitative trading strategies. This module, referred to as the Performance Trader, is specifically designed to backtest, optimise, and live-trade multi-condition quantitative strategies in diverse market scenarios. Leveraging the foundational high-performance compute engine of the Invesdwin platform, this component has proven instrumental in achieving robust trading performance.

The Performance Trader is used to optimise existing trading strategies to deliver an average annual growth rate (AAGR) of over 20%, outperforming traditional benchmarks in both profitability and risk-adjusted returns. The ability to quickly iterate and refine strategies in the face of dynamic market conditions is a key advantage provided by this tool. The platform's high-speed backtesting capabilities, combined with the flexibility of the code base, significantly reduced the time required for evaluating complex multi-condition strategies. This reduction in backtesting times, by an order of magnitude compared to traditional tools, has enabled portfolio managers to focus on strategy refinement and robustness analysis.

In live trading scenarios, the Performance Trader utilises the low-latency features of the underlying platform to execute decisions in real-time. Its integration with custom risk management modules allows seamless adaptation to changing market regimes. Furthermore, the module benefits from an advanced monitoring and logging system, ensuring transparent decision-making.

By combining real-time decision-making with the computational power of the Invesdwin platform, 123 Invest Group has established the Performance Trader as a cornerstone of their trading operations. The Performance Trader not only streamlines the development and deployment of strategies but also ensures their scalability and adaptability in both backtesting and live environments.

VIII. Summary

Based on our results, the Invesdwin platform is the fastest publicly available platform for testing trading strategies, outperforming the fastest contender by almost an order of magnitude. Detailed performance results are presented in an upcoming paper [2]. This level of computational performance is achieved by a range of implementation techniques, most notably in-memory usage of (compressed) data structures and a tuned NoSQL implementation of the underlying database of market data. Additionally, the platform features a powerful generator of trading strategies, based on a formal expression language as a domain-specific language. Combining both components opens the door to fully automated strategy generation and evaluation. A current focus in this direction is the robustness analysis of automatically generated strategies. Preliminary results show the benefits of certain hybrid techniques that combine the strategy generation, based on Differential Evolution, with domain-specific ML techniques such as SFA and DEA for portfolio selection. These case studies demonstrate the value of the platform as a research tool in the area of FinTech.

The technical basis of the Invesdwin platform, encompassing high-performance libraries for its compute engine, is available publicly under the LGPL open-source license here: “invesdwin.de”. The trading platform with the compute engine is available to researchers free of charge on an invitation basis under a community license with full

code access. This decision has been made in the hope of growing a powerful FinTech framework, similar in spirit to the ImageJ platform for image progressing or the Weka framework for general ML technologies. Interested researchers and potential contributors to this effort are encouraged to contact the author of the platform directly to gain access to the full Invesdwin platform via E-Mail: “edwinstang@gmail.com”.

About the Author

Dr Hans-Wolfgang Loidl is Associate Professor in Computer Science at Heriot-Watt University, Edinburgh, Scotland. His main research areas are programming languages, high-performance computing, and formal foundations of computing. He has developed system-level technologies for various domains, with a current focus on FinTech, with its specific requirements on high-performance processing of large data sets and the increasing interest in applying data-driven machine-learning techniques. He has been working with the developer of the Invesdwin platform since 2020, acting as primary supervisor to his DBA studies. In that capacity, Dr Loidl obtained in-depth knowledge about the Invesdwin platform, its performance aspects, and its internal software architecture. While this platform remains a key tool for continuing joint research, Dr Loidl is not involved in the business side of the use of this platform that is currently handled through 123 Invest Group at: “1-2-3-invest.de”.

References

[1] “Differential Evolution”, K. Price et al, Springer, 2014. ISBN: 978-3-642-42416-8.

[2] Edwin Stang, Hans-Wolfgang Loidl, Boulis M. Ibrahim “High-performance Machine Learning for FinTech”, submitted to Journal of Parallel and Distributed Computing, July 2024.

[3] Edwin Stang, Hans-Wolfgang Loidl, Boulis Ibrahim. “A Simple Expression Language for Automating Strategy Development Processes.”, March 2022. DOI: 10.13140/RG.2.2.18081.48484.