



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ZARZĄDZANIA

KATEDRA Informatyki Biznesowej i Inżynierii Zarządzania

Projekt dyplomowy

Teoretyczne i praktyczno - informatyczne aspekty handlu algorytmicznego

Theoretical and Practical IT Aspects of Algorithmic Trading

Autor: Bartłomiej Jamiołkowski

Kierunek studiów: Informatyka i Ekonometria

Opiekun projektu: dr Beata Basiura

Kraków, 2022

Spis treści

Wstęp	3
1. Wprowadzenie do handlu algorytmicznego.....	4
1.1 Pojęcie i rozwój historyczny handlu algorytmicznego.....	4
1.2 Problematyka automatyzacji handlu	6
1.2.1 Opis wybranych wyzwań.....	6
1.2.2 Zalety stosowania algorytmów w handlu	7
1.2.3 Podstawowe wady systemów	8
1.3 Znane platformy handlu algorytmicznego	8
1.3.1 QuantConnect	9
1.3.2 TradeStation.....	10
1.3.3 AmiBroker	10
2. Wybrane komponenty systemu transakcyjnego.....	12
2.1 Market Data Adapter	15
2.2 Complex Event Processing Engine	16
2.3 Order Management System.....	18
3. Wybrane strategie handlu algorytmicznego.....	19
3.1 Trend following.....	20
3.1.1 Exponentially Weighted Moving Average Crossover.....	21
3.2 Mean Reversion.....	23
3.2.1 Bollinger Band.....	24
3.3 Statistical Arbitrage.....	26
3.3.1 Cross Market Arbitrage	27
4. Podsumowanie	32
Literatura.....	33
Źródła internetowe	35
Spis rysunków	36
Spis tabel	37

Wstęp

Motywacją stojącą za wyborem handlu algorytmicznego jako tematu pracy jest pragnienie zrozumienia tej formy inwestowania oraz elementów informatyki umożliwiających zastosowanie go w praktyce. Zainteresowanie algorytmami w obrocie giełdowym pojawia się w momencie przeglądania literatury dedykowanej nowoczesnym technikom programowania finansowego. W świetle postępującej rewolucji informacyjnej w tym obszarze za właściwe uznaje się pogłębienie wiedzy o handlu algorytmicznym.

Przedmiot rozważań stanowi informatyczny proces realizacji kupna i sprzedaży instrumentów finansowych. Obejmuje on swoim zasięgiem: teorię handlu algorytmicznego, opis najważniejszych komponentów zautomatyzowanego systemu transakcyjnego oraz wybrane strategie handlu.

Celem pracy jest zaprezentowanie czytelnikowi wybranych narzędzi informatycznych, które stanowią fundament handlu algorytmicznego.

Praca składa się z trzech rozdziałów, wstępu i podsumowania. Pierwszy z nich stanowi wprowadzenie czytelnika do tematyki handlu algorytmicznego. Zawiera definicję tego handlu oraz analizę tego terminu popartą historią jego powstania. W dalszej części rozdział porusza problematykę automatyzacji handlu wynikającą z jego nieustannego rozwoju. Prezentowane są: wybrane wyzwania natury technicznej, zalety oraz wady. Ostatnim elementem tego rozdziału jest przedstawienie praktycznych zastosowań handlu algorytmicznego na przykładzie wybranych platform transakcyjnych.

Kontynuację przyjętej retoryki stanowi rozdział drugi opisujący konceptualny system transakcyjny oparty na przetwarzaniu zdarzeń. Bazując na jego architekturze omawiane są w nim najważniejsze wybrane komponenty potrzebne do realizacji transakcji.

Rozdział trzeci zaznajamia czytelnika z wybranymi strategiami handlu, które są wdrażane w systemach tego typu. Na kilku przykładowych inwestycjach pokazane zostały wyniki działania wybranych strategii.

Źródłami wykorzystywanymi w projekcie są: książki, artykuły, czasopisma i strony internetowe. Obejmują one swoim zakresem: tematykę handlu algorytmicznego, architektury systemów oraz programowania finansowego.

1. Wprowadzenie do handlu algorytmicznego

Handel algorytmiczny jest rozwijającą się dziedziną nie posiadającą jednoznacznej definicji. Z tego względu w rozdziale tym zostaje podjęta próba sformułowania opisu tej formy inwestowania. W ramach tego analizowane są najważniejsze wydarzenia historyczne. Historyczne podejście do rozwoju informatyki handlu algorytmicznego stanowi uzasadnienie dla obranej retoryki.

Oprócz tego w niniejszym rozdziale opisywane są wybrane: problemy techniczne tego rodzaju inwestycji, zalety oraz wady mające wpływ na przebieg tworzenia systemów transakcyjnych.

Ostatnim zagadnieniem w tym rozdziale jest prezentacja trzech wybranych platform umożliwiających zastosowanie handlu algorytmicznego w praktyce.

1.1 Pojęcie i rozwój historyczny handlu algorytmicznego

Opierając się na przeprowadzonych badaniach literaturowych można spróbować sformułować ogólną opinię, że istotą handlu algorytmicznego jest zastosowanie zaprogramowanych instrukcji w obrocie instrumentami finansowymi. W szczególnym znaczeniu mowa tu o systemach oraz strategiach handlu. Połączone razem wspomagają lub zastępują one człowieka w przyspieszonym kupowaniu i zbywaniu niektórych aktywów finansowych.

Przykładem naukowej interpretacji tego terminu jest artykuł¹ autorstwa: Philipa Treleavena, Michala Galasa i Vidhi Lalchanda. Autorzy definiują handel algorytmiczny jako formę handlu wykorzystującą wyszukane algorytmy (zaprogramowane systemy) do automatyzacji całości lub części procesu obrotu instrumentami finansowymi. Według uczonych obejmuje on: uczenie, dynamiczne planowanie, wnioskowanie i podejmowanie decyzji.

Bez wątpienia przytoczona publikacja słusznie zalicza systemy transakcyjne do obszaru objętego terminem handel algorytmiczny. Badacze zwracają uwagę na istnienie półautomatycznych oraz w pełni zautomatyzowanych systemów i strategii pozwalających automatycznie przeprowadzać transakcje kupna/sprzedaży. Oprócz tego wskazują jakie

¹ Treleaven P., Lalchand V., Galas M., Algorithmic Trading Review, „Communications of the ACM” 2013, Volume 56, s. 76

procesy mogą być w nich realizowane. Niemniej jednak zbyt ogólnie definiują cel ich stosowania jakim jest przyspieszenie szybkości oraz częstotliwości transakcji handlu.

Należy przyjąć, że kluczem do zrozumienia tego celu oraz nazewnictwa handlu algorytmicznego jest analiza ciągu istotnych wydarzeń z XX i XXI wieku. Giełdy w czasach poprzedzających rewolucję informacyjną charakteryzują się obecnością ludzi na parkietach. Wraz z postępem technologicznym w pewnym momencie pojawiają się elektroniczne systemy, które zmieniają dotychczasowy porządek.

Opisuje to książka² napisana przez Raje Velu wraz z Maxencem Hardym i Danielem Nehrenem, w której przełomową datą w tym temacie jest rok 1969. To właśnie wtedy *Institutional Networks Corporation* uruchamia skomputeryzowane łącze *Instinet*. Stanowi połączenie między: bankami, funduszami inwestycyjnymi oraz ubezpieczycielami umożliwiające im szybszy handel pomiędzy sobą z pominięciem giełdy *NYSE*. Badacze uznają *Instinet* za pierwszy przykład *Electronic Communication Network (ECN)* będącego alternatywnym podejściem do obrotu giełdowego. Uczeń podają, że w latach 80 i 90 *ECN* zyskuje na popularności ze względu na pojawienie się większej liczby miejsc wykorzystujących to podejście. Jako przykład przytaczają firmy *Archipelago* i *Island ECN*. Skutkiem tych zmian jest powstanie trendu nazywanego fragmentacją płynności. Według autorów przejawia się on poprzez rozproszenie zainteresowania papierami wartościowymi. Stanowi to powód do elektronizacji handlu.

Powyższy fragment tekstu opisuje wydarzenia, które stworzyły podwaliny handlu algorytmicznego. W ostatnich dekadach XX wieku skomputeryzowany handel nazywany był *Electronic Trading*, którego przykładem jest *ECN*.

Według strony³ *ECN* stanowi skomputeryzowany system, który automatycznie dopasowuje zlecenia kupna i sprzedaży papierów wartościowych na rynku. Handel *ECN* jest szczególnie pomocny, gdy inwestorzy z różnych obszarów geograficznych chcą sfinalizować bezpieczną transakcję bez użycia strony trzeciej.

Istnieją odmienne poglądy co do faktycznego momentu wyłonienia się handlu algorytmicznego. Niektóre źródła wskazują na lata 80 XX wieku, podczas gdy inne powołują się na reformę decymalizacji w USA z 2001 roku.

W projekcie przyjmuje się pierwszą propozycję z racji tego, że wspomniana reforma jest odpowiedzialna jedynie za wzrost użycia handlu algorytmicznego.

² Velu R., Hardy M., Nehren D., *Algorithmic Trading and Quantitative Strategies*, CRC Press, Abingdon 2020, s. 4-5

³ Hayes A., *Electronic Communication Network*, <https://www.investopedia.com/terms/e/ecn.asp>, dostęp 08.062022

Publikacją godną uwagi jest książka⁴ autorstwa Edwarda Leshika i Jane Cralle. Autorzy za moment powstania handlu algorytmicznego uznają rok 1980. W tamtym czasie powstał zespół naukowców banku *Morgan Stanley*, który opracował strategię handlu parami. Wspomniana grupa zrodziła wiele osób, które były pionierami intensywnego wykorzystywania mocy obliczeniowej w celu uzyskania przewagi konkurencyjnej nad kolegami. Gdy moc komputerów wzrosła zgodnie z prawem Moore'a (prędkość podwaja się co osiemnaście miesięcy), siła skomputeryzowanego handlu algorytmicznego była nieodparta. Ten postęp został połączony z wynalezieniem bezpośredniego dostępu do rynku dla członków niebędących członkami giełdy. Umożliwił im on dokonywanie transakcji przez indywidualnych handlowców za pośrednictwem ich domów maklerskich. Wkrótce wszystkie główne biura handlowe uruchomiły algorytmy.

Analizując zaprezentowane dzieła można zrozumieć realia, które wpływają na pojawienie się handlu algorytmicznego. Potwierdzają one sformułowaną definicję oraz uzasadniają przyspieszanie obliczeń w tej formie handlu.

1.2 Problematyka automatyzacji handlu

Nieustanny rozwój handlu algorytmicznego powoduje powstanie nowych korzyści i wyzwań natury technicznej. Poznanie ich stanowi podstawę pisania algorytmów oraz projektowania systemów w taki sposób by działały efektywnie. Do najistotniejszych wyzwań zalicza się integrację danych i automatyczną kontrolę systemów transakcyjnych. Swoją tematyką obejmują początek oraz koniec procesu realizacji transakcji.

1.2.1 Opis wybranych wyzwań

Pierwszy problem przedstawia w swojej książce⁵ Kendall Kim zwracając uwagę na wyzwanie przed jakim stoi projektant systemu pragnący zaimplementować integrację danych. Według autora brak przesyłanych codziennie danych statycznych powodowałby niemożność dokonania odpowiedniej wyceny instrumentu finansowego oraz jego uchwycenia w systemie.

⁴ Leshik E., Cralle J., *An Introduction to Algorithmic Trading: Basic to Advanced Strategies*, Wiley 2011, s. 7

⁵ Kim K., *Electronic and Trading Technology: the Complete Guide*, Academic Press, Cambridge 2007, s. 184

Problem ten związany jest z parametryzacją systemu, która określa w jaki sposób różne: elementy danych, instrumenty i kontrahenci są mapowani do modelu danych platformy transakcyjnej. Za największe wyzwanie związane z parametryzacją badacz uznaje trudność w modyfikowaniu interfejsu po przejściu systemów do produkcji. Nie wystarczy tylko zidentyfikować systemy źródłowe wymaganych danych i napisać skrypty przesyłania. Należy się również zająć: jakością danych, czyszczeniem oraz sposobem ich przesyłania.

Drugie wyzwanie opisuje w czasopiśmie⁶ Jackie Shen. W zamieszczonym tekście uczony wyjaśnia czytelnikowi dlaczego kontrola systemów handlu algorytmicznego jest tak ważna. Według niego: awarie, incydenty operacyjne oraz wadliwy projekt mają negatywny wpływ na ceny papierów wartościowych, główne poziomy indeksów lub powiązane instrumenty pochodne. Wymienione sytuacje są ryzykowne dla handlarza. Spośród występujących zagrożeń badacz wymienia ryzyko operacyjne i technologiczne. Polega ono na wadliwym monitorowaniu: systemów transakcyjnych, sieci lub baz danych. Ze względu na autonomiczny charakter większość zachowań algorytmów musi być kontrolowana w sposób zautomatyzowany lub niskodotykowy.

Kontrole mogą być proste dla małych firm handlowych, które koncentrują się na bardzo ograniczonym zestawie papierów wartościowych przy użyciu ograniczonego zestawu algorytmów. W przypadku większych podmiotów takich jak: bank inwestycyjny, broker-dealer lub firma handlowa to zadanie stanowi wyzwanie. Uczony uważa, że zniechęcającym zadaniem jest opracowanie rygorystycznych i skutecznych ram kontroli, które są przejrzyste i podlegające audytowi.

1.2.2 Zalety stosowania algorytmów w handlu

Mimo poważnych wyzwań związanych ze stosowaniem handlu algorytmicznego wynikające z jego stosowania korzyści są na tyle duże, że motywują inwestorów do implementacji tej formy handlu. Najistotniejsze z nich opisują w swojej książce⁷ Sebastien Donadio i Sourav Ghosh. Autorzy w zamieszczonym tekście wymieniają zalety takie jak:

⁶ Shen J., Nine Challenges in Modern Algorithmic Trading and Controls, „Algorithmic Trading and Controls” 2021, Nr 1, s. 6-7

⁷ Donadio S., Ghosh S., Learn Algorithmic Trading, Pact, Birmingham 2019, s. 19

- wyjątkowa skuteczność komputerów w wykonywaniu powtarzalnych zadań opartych na regułach;
- systemy mogą niezwykle szybko wykonywać instrukcje i algorytmy;
- można je bezproblemowo skalować i wdrażać w wielu instrumentach finansowych;
- algorytmy nie mają emocji, więc nie odbiegają od tego, do czego są zaprogramowane (przy założeniu ich poprawnej implementacji).

1.2.3 Podstawowe wady systemów

Niezaprzeczalnie rozwiązania handlu algorytmicznego przynoszą nie tylko korzyści, ale również posiadają wady.

Według książki⁸ autorstwa Michaela L. Hallsa-Moorea istnieją dwie główne przeciwwskazania stosowania tej formy handlu. Specjalista wymienia negatywne aspekty takie jak:

- handel algorytmiczny wymaga wyższej bazy kapitałowej, ze względu na wymogi stawiane przez pośredników i regulatorów;
- pozyskiwanie strumieni danych do strategii jest często związane z kosztami, ponieważ dane dobrej jakości są dystrybuowane przez wyspecjalizowane podmioty;
- osoba posługująca się handlem algorytmicznym musi być stosunkowo biegła zarówno w programowaniu jak i w modelowaniu naukowym, co zawęża grono użytkowników.

1.3 Znane platformy handlu algorytmicznego

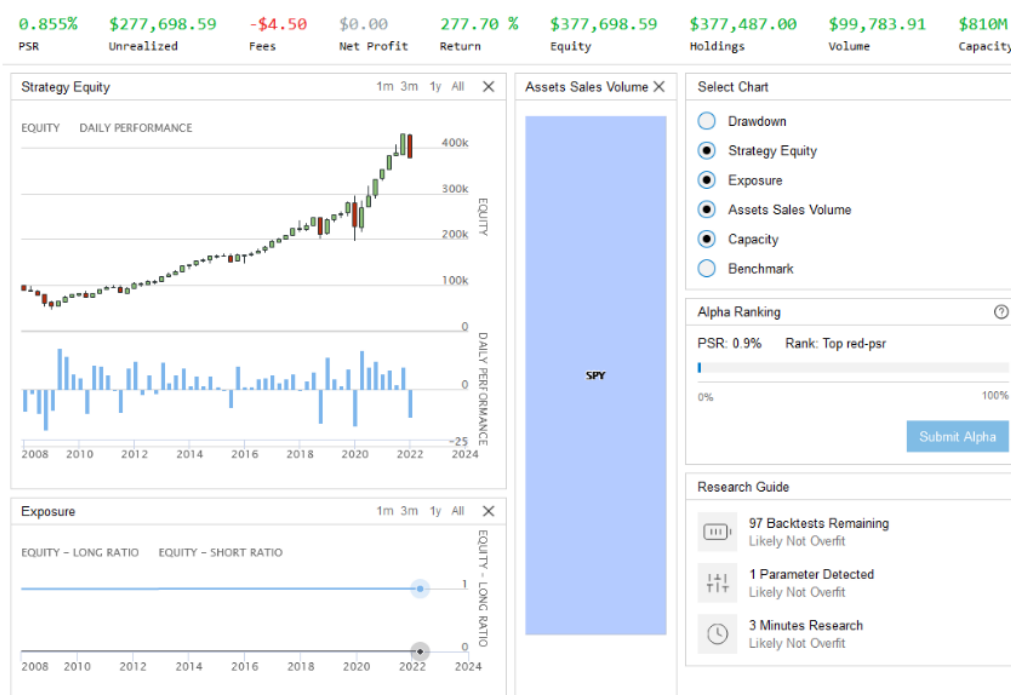
Obserwując obszar handlu algorytmicznego można zauważyć liczne podmioty operujące w tej dziedzinie. Cechują się one różnym zakresem usług. W tej części rozdziału zaprezentowane zostaną wybrane popularne platformy, które udostępniają niektóre informatyczne aspekty swoich systemów.

⁸ Halls-Moore L. M., Successful Algorithmic Trading, 2015, s. 8

1.3.1 QuantConnect

QuantConnect to platforma chmurowa typu *open-source* dedykowana handlowi algorytmicznemu. Swoje działanie opiera na stworzonym systemie *LEAN* wzorowanym na podejściu *stream processing*. Świadczy ona usługi z zakresu: badań, testowania historycznego strategii oraz pośredniczy w handlu na żywo. Docelowymi obsługiwanyymi językami programowania są: *Python*, *F#* oraz *C#*. Wykorzystuje się je do pisania własnych strategii w narzędziu *Lab/Terminal* oraz komunikacji z *frameworkami* platformy. Umożliwia to użytkownikowi na przykład pozyskiwanie danych historycznych lub symulację działania stworzonego algorytmu handlu.

Pomiary działania strategii są przeprowadzane w narzędziu platformy *QuantConnect* o nazwie *Backtesting*, którego rezultaty działania prezentuje rysunek 1.



Rys. 1. Strona wyników Backtestingu platformy QuantConnect
Źródło: [20]

Pokazana na rysunku 1 strona z wynikami wyświetla informacje takie jak: notowania cen akcji instrumentu finansowego, statystyki oraz wydajność algorytmu.

1.3.2 TradeStation

TradeStation to elektroniczna platforma typu *closed-source* używana do handlu instrumentami finansowymi. Swoim klientom zapewnia gotowe: strategie, komponenty, zbiory danych oraz możliwość testowania historycznego. To co wyróżnia *TradeStation* spośród konkurencji to własny zastrzeżony język programowania *EasyLanguage*. Cechuje go duża liczba: prostych słów, funkcji oraz właściwości. Umożliwiają one programistom tworzenie narzędzi takich jak: wskaźniki, strategie lub aplikacje handlu.

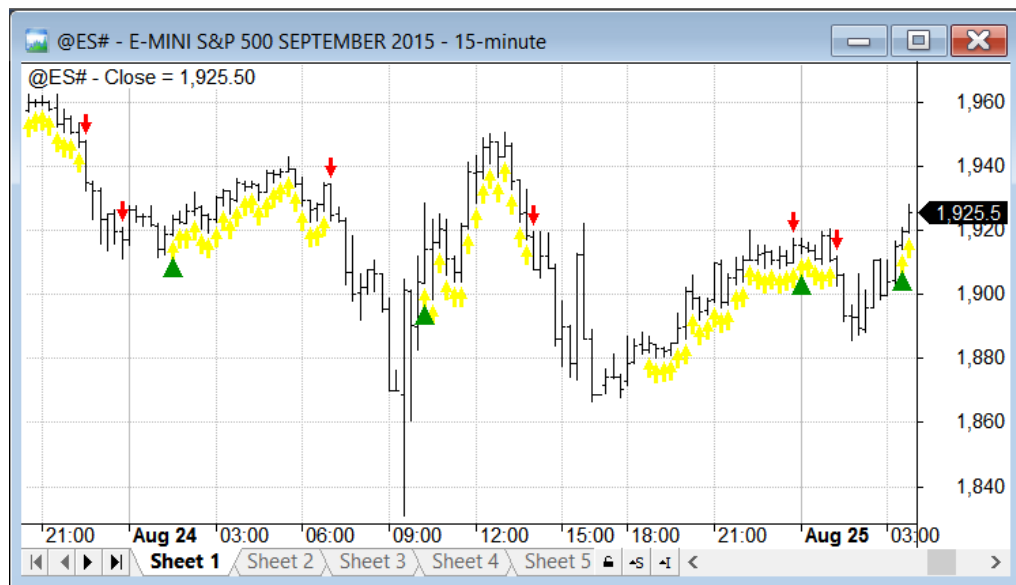
Platforma posiada również rozbudowane narzędzia do tworzenia wykresów. Przy ich wykorzystaniu można zwizualizować na przykład linię trendu widoczną na rysunku 2.



Rys. 2. Ceny akcji instrumentu finansowego z naniesioną linią trendu na platformie TradeStation
Źródło: [21]

1.3.3 AmiBroker

AmiBroker to analityczna platforma oferująca użytkownikom liczne funkcje w handlu instrumentami finansowymi. Najpopularniejszą z nich jest tworzenie wykresów cechujące się wbudowanymi wskaźnikami oraz szerokim przedziałem czasowym. Przykładem takiego wykresu jest zamieszczona na rysunku 3 wizualizacja.



Rys. 3. Wyniki testowania działania strategii na platformie AmiBroker
Źródło: [25]

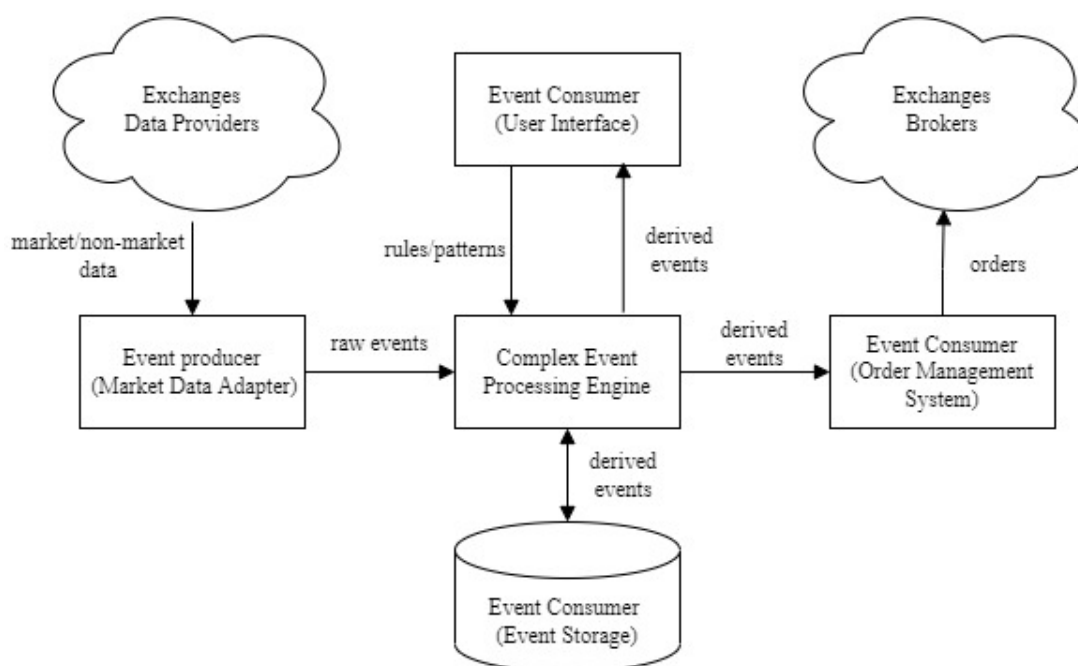
Rysunek 3 prezentuje zmieniające się ceny zamknięcia wybranego instrumentu finansowego z naniesionymi sygnałami kupna i sprzedaży. Naniesienie oznaczeń ma na celu na przykład regulację liczby dziennych transakcji w *backtestingu*.

Innym aspektem platformy jest *AmiBroker Formula Editor (AFL)*, który jest wykorzystywany do tworzenia nowych strategii. Ułatwia użytkownikom pisanie algorytmów handlu poprzez podświetlenie elementów składni takich jak nazwy zmiennych lub funkcje.

2. Wybrane komponenty systemu transakcyjnego

Systemy handlu algorytmicznego są zróżnicowane pod względem architektury w zależności od potrzeb ich właścicieli. Nie mniej jednak istnieją uniwersalne zasady, którymi należy się kierować podczas ich tworzenia.

Niniejszy rozdział ma na celu zapoznanie czytelnika z wybranymi komponentami zautomatyzowanego systemu transakcyjnego. Reprezentują one nowoczesną technologię przetwarzania zdarzeń znaną jako *Complex Event Processing (CEP)*. Konceptualny model systemu tego typu został pokazany na rysunku 4.



Rys. 4. Konceptualny model systemu handlu algorytmicznego
Źródło: Opracowanie własne na podstawie [5]

Zaproponowany na rysunku 4 schemat ukazuje uniwersalne komponenty systemu połączone strzałkami reprezentującymi kierunki przepływu danych. Najważniejsze elementy znajdują się na osi poziomej. Pierwszy z nich to *Market Data Adapter*, który komunikuje się ze światem zewnętrznym w celu pozyskiwania danych. Rezultatem jego działania są wychodzące strumienie zdarzeń do komponentu *Complex Event Processing Engine*. Element ten jest połączony z *User Interface* oraz *Event Storage*. Wspomniany interfejs dostarcza reguł użytkownika do *CEP Engine* i prezentuje wyniki pracy silnika. *Event Storage* jest miejscem przechowywania danych wykorzystywanych do wtórnego przetworzenia przez silnik. Efektem pracy *CEP Engine* jest nowy strumień

danych, który trafia do *Order Management System*. Stamtąd zlecenia handlu są przesyłane do odbiorców na przykład brokerów lub bezpośrednio na giełdę.

Powyższa interpretacja stanowi uproszczony opis procesu handlu algorytmicznego z perspektywy praktyczno-informatycznej. Zrozumienie postaci przedstawionego modelu oraz zachodzących w nim złożonych procesów wymaga analizy pojęcia *Complex Event Processing*.

Pojęcie to definiuje w swoim artykule⁹ Sylvain Hallé, który opisuje zdarzenia jako sekwencje jednostek danych. Przetwarza się je poprzez wykorzystanie narzędzi i technik zaprojektowanych w celu efektywnego procesowania zapytań. Jako przykład przytacza on: ruchome okna, funkcje agregacji oraz proste sekwencyjne wzorce w postaci krotek.

Oprócz tego uczony wskazuje na najistotniejszy aspekt *CEP* jakim jest kompleksowość. Przejawia się ona w możliwości wykonywania korelacji pomiędzy zdarzeniami z tego samego strumienia lub zdarzeniami pochodzącymi z różnych strumieni. Umożliwia to tworzenie zagregowanych zdarzeń na podstawie zdarzeń niższego poziomu. Mogą one być wykonywane kaskadowo, gdzie strumienie wyjściowe poprzedniego procesu stają się strumieniami wejściowymi nowego procesu.

Badacz zwraca również uwagę na wyzwanie przed jakim stoi użytkownik pragnący stworzyć system tego typu. *Sylvain Hallé* przedstawia dwa warianty postępowania w takiej sytuacji. Pierwszym z nich jest zaprojektowanie systemu zgodnego z teorią *CEP* akceptując wszystkie jego niedoskonałości. Drugą opcję stanowi zaprogramowanie własnej wersji systemu przy wykorzystaniu języków skryptowych niskiego poziomu. Autor zaznacza, że taki system powinien posiadać: zrozumiały model obliczeniowy, wyższą abstrakcję oraz rozsądną wydajność.

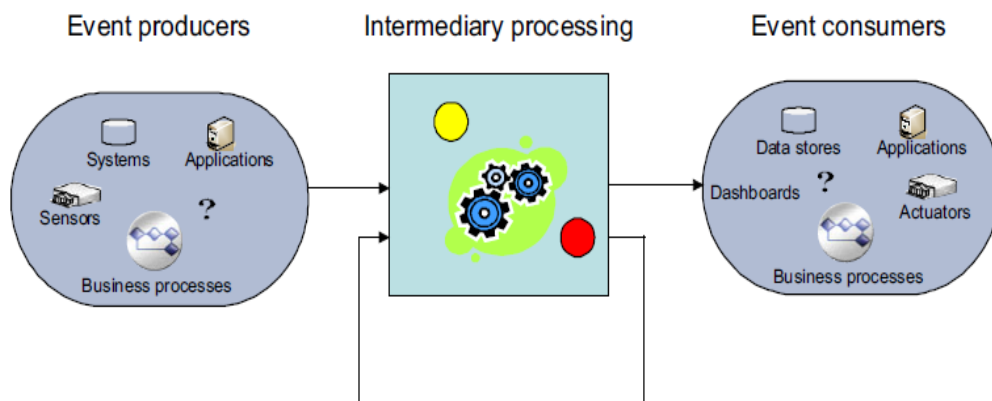
Przytoczony artykuł stanowi ważne źródło informacji o *CEP*. Badacz w swojej wypowiedzi wyjaśnia egzystencję niestandardowych systemów dedykowanych na przykład handlu algorytmicznemu. Oprócz tego ukazuje on możliwe przepływy zdarzeń takich jak powtórne wykorzystanie strumieni danych, które jest widoczne na grafice między *CEP Engine* a *Event Storage*. Nie mniej jednak publikacja nie skupia się na elementach tworzących standardowy system *CEP*.

Uzupełnienie tego tematu stanowi książka¹⁰ autorstwa Ophera Etziona i Petera Nibleta. W swoim dziele uczeni pokazują uproszczoną architekturę systemu przetwarzania

⁹ Hallé S., *From Complex Event Processing to Simple Event Processing*, 2017, s. 1-2

¹⁰ Etzion O., Niblet P., *Event Processing in Action 1st Edition*, Manning Publications Co., Stamford 2011, s. 40-42

zdarzeń. Do jej zobrazowania badacze wykorzystują monolityczne komponenty powiązane strzałkami (strumieniami zdarzeń). W podanej publikacji informują, że bloki symbolizują bardziej złożoną strukturę systemu. Omawiane aspekty są prezentowane wizualnie na rysunku 5.



Rys. 5: Uproszczona architektura systemu przetwarzania strumieni zdarzeń
Źródło: Opracowano na podstawie [5]

Po lewej stronie grafiki przedstawionej na rysunku 5 zauważalny jest blok o nazwie *event producers* reprezentujący niezidentyfikowaną ilość komponentów generujących zdarzenia. Według autorów istnieje szeroki zakres kształtów i rozmiarów producentów zdarzeń takich jak: czujniki sprzętowe, bity oprzyrządowania programowego lub fragmenty logiki oprogramowania aplikacji.

Po prawej stronie na tym samym rysunku znajduje się blok oznaczony jako *event consumers*. Symbolizuje on komponenty oddziałujące na odbierane zdarzenia. Mogą one: przechowywać zdarzenia, wyświetlać je w interfejsie użytkownika lub podejmować inne działania.

Ostatnim rozważanym elementem wizualizacji z rysunku 5 jest blok *intermediary processing*. Według uczonych zawiera on w sobie sekwencje podkomponentów. Przetwarzają one surowe strumienie (z ang. *raw events*) emitowane przez producentów a następnie generują zdarzenia pochodne (z ang. *derived events*). Mogą one być ponownie procesowane przez te komponenty lub trafić do konsumentów.

Analizując wspomnianą publikację [4] nie ma wątpliwości, że badacze opisują podstawy technologii *Complex Event Processing*. Przyjmując ich retorykę, za producenta zdarzeń w modelu systemu handlu algorytmicznego należałoby uznać komponent *Market Data Adapter*. W takim przypadku wychodzące od niego strumienie zdarzeń określane by

były jako surowe (z ang. *raw events*). Z drugiej strony konsumentami zdarzeń pochodnych (z ang. *derived events*) byłoby: *User Interface*, *Event Storage* oraz *Order Routing System*.

2.1 Market Data Adapter

Niezaprzeczalnie podstawą każdego efektywnie działającego systemu handlu algorytmicznego są dostarczone dane.

Opisują to w swoim artykule¹¹: Feng Wang, Keren Dong oraz Xiaotie Deng proponując podział danych ze względu na źródło ich pochodzenia i czas. Według autorów dane do systemów handlu algorytmicznego pozyskuje się z rynku papierów wartościowych lub newsów internetowych. Są to zarówno dane rzeczywiste jak i historyczne wykorzystywane odpowiednio w analizie bieżącej sytuacji rynkowej oraz do treningu modeli strategii handlu.

Najprawdopodobniej za omawiany przez badaczy proces odpowiedzialny jest *Market Data Adapter*. Stanowi on odmianę tak zwanego adaptera ukierunkowaną na pozyskiwanie danych związanych z rynkiem papierów wartościowych.

Według książki¹² adapter jest zaliczany do producentów zdarzeń. Jego praca polega na nie wykrywaniu bezpośrednio zdarzeń, ale zbieraniu dostępnych informacji z innych źródeł. Wykorzystuje on je do generowania zdarzeń. Autorzy przedstawiają działania adaptera na przykładzie czujników. Wskazują oni, że pojedynczy adapter może być używany do podłączenia wielu czujników do sieci i w ten sposób działać jako brama koncentracyjna. Co istotniejsze ten komponent może przetłumaczyć protokół używany przez czujnik na inny protokół.

W kontekście handlu algorytmicznego *Market Data Adapter* stanowi bramę koncentracyjną dla danych z zewnętrznych źródeł takich jak giełdy, czy wyspecjalizowane podmioty. Do swojego funkcjonowania wykorzystuje protokół *FIX* (z ang. *Financial Information eXchange*). Według strony¹³ jest to niezastrzeżony, otwarty i bezpłatny standard stworzony w ramach współpracy pomiędzy instytucjami finansowymi. Powszechnie używa się go na całym świecie do realizacji transakcji.

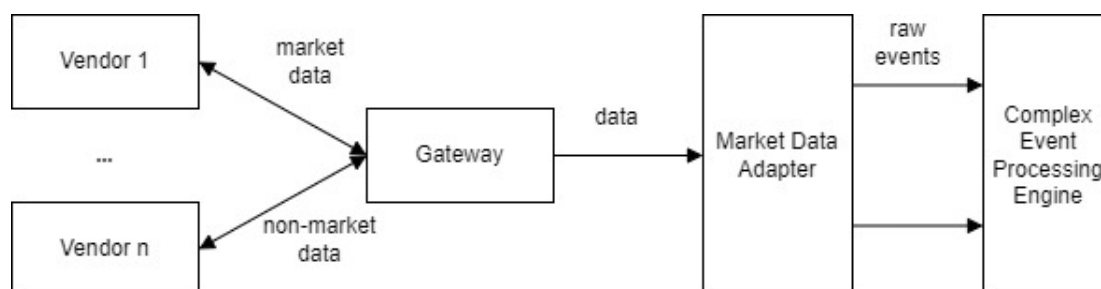
¹¹ Wang F., Dong K., Deng X., Algorithmic trading system: design and applications, Frontiers of Computer Science in China 2009, Volume 3, s. 236-237

¹² Etzion O., Nibblet P., Event Processing in Action 1st Edition, Manning Publications Co., Stamford 2011, s. 93

¹³ What is FIX?, <https://www.fixtrading.org/what-is-fix/>, dostęp 07.06.2022

Przepływ danych przez *Market Data Adapter* ukazuje poniższy schemat z rysunku

6.



Rys. 6. Model działania komponentu Market Data Adapter

Źródło: Opracowanie własne na podstawie [1]

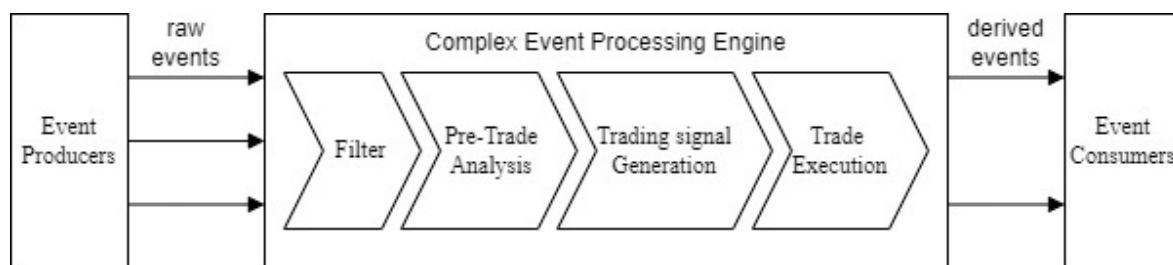
Na rysunku 6 po lewej stronie widoczni są dostawcy danych. Komunikują się z nimi bramki, które wysyłają zapytania i otrzymują od nich odpowiedzi. Pozyskane dane trafiają do *Market Data Adaptera* a nowe strumienie zdarzeń są przekierowywane do *CEP Engine*.

2.2 Complex Event Processing Engine

Według artykułu¹⁴ autorstwa: Dong Wanga, Mingquan Zhoua, Sajida Aliego, Pengbo Zhou, Yusong Liu i Xuesong Wanga *Complex Event Processing Engine* jest najważniejszym komponentem systemu wykorzystującego tę technologię. Badacze w swojej pracy opisują działanie silnika za pomocą trzech etapów. Pierwszy krok stanowi przechwycenie strumienia zdarzeń przez silnik. Kontynuację tego procesu stanowi analiza i detekcja znaczących sytuacji za pomocą zaimplementowanych reguł. Ostatnim etapem jest podjęcie natychmiastowych działań przez *CEP Engine*.

W handlu algorytmicznym przedstawiony proces może różnić się w zależności od rodzaju strategii, które wykorzystuje system. Analizowane źródła na ten temat najczęściej podają etapy widoczne na rysunku 7.

¹⁴ Wang D., Zhou W., Ali S., Zhou P., Liu Y., Wang X., A Novel Complex Event Processing Engine for Intelligent Data Analysis in Integrated Information Systems, International Journal of Distributed Sensors Network, 2016, s. 2



Rys. 7. Complex Event Processing Engine w modelu systemu handlu algorytmicznego
Źródło: Opracowanie własne na podstawie [15]

Przedstawiony silnik na rysunku 7 realizuje proces składający się z następujących etapów: *Filter*, *Pre-Trade Analysis*, *Trading Signal Generation* i *Trade Execution*. Filtr symbolizuje proces pobierania zdarzeń na temat określonego przez użytkownika instrumentu finansowego. Pozostałe etapy szerzej opisuje dostępna literatura.

Przykładem publikacji na ten temat jest artykuł¹⁵ autorstwa *Davea Cliffa* i *Philipa Trelavena*. Badacze dzielą w nim process handlu algorytmicznego na trzy fazy: *pre-trade analysis*, *signal generation* i *trade order execution*.

Pierwszy etap składa się z trzech podkomponentów: *alpha model* (predykcja zachowania instrumentów finansowych), *risk model* (ewaluacja potencjalnych strat) oraz *cost model* (kalkulacja potencjalnego kosztu handlu).

Faza *signal generation* zawiera podkomponent *portfolio construction model*, który przyjmuje dane od modeli z pierwszego etapu. Następnie wybiera najlepsze portfolio instrumentów finansowych do handlu kierując się maksymalizacją zysku i redukcją kosztów oraz ryzyka.

Ostatnia fazą według uczonych jest *order execution*, w czasie której przyjmowane są zlecenia handlu i inne dane wykorzystywane do przeprowadzenia egzekucji w najefektywniejszy sposób.

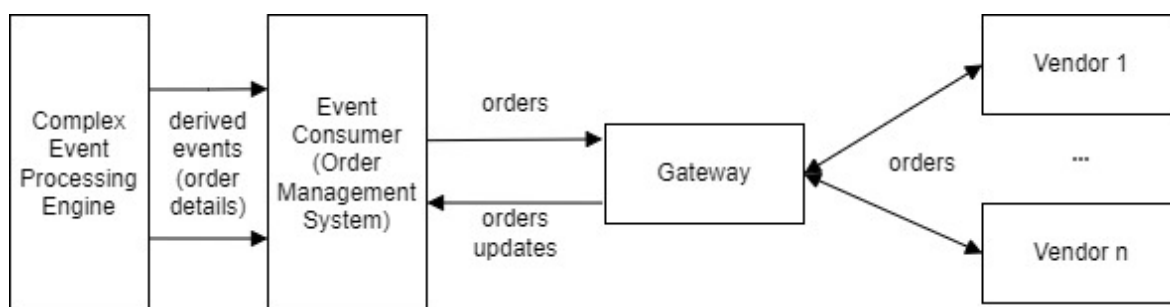
Szczególne uwagę należy zwrócić na *alpha model*, ponieważ to w nim implementowane są strategie handlu mające wpływ na pojawienie się sygnału kupna i sprzedaży.

¹⁵ Cliff D., Trelaven P., *Technology Trends in the Financial Markets: A 2020 Vision*, 2010, s. 11-12

2.3 Order Management System

Ostatnim omawianym elementem modelu systemu handlu algorytmicznego jest *Order Management System*. Według książki¹⁶ *OMS* stanowi komponent kolekcjonujący zlecenia handlu. Do jego zadań należy: stworzenie, egzekucja, uaktualnienie, anulowanie oraz odrzucenie zlecenia. Na początku procesu analizuje on dostarczone zdarzenia pod względem poprawności. Jeśli występują błędy w danych potencjalne zlecenia nie są procedowane, a informacje o tych zdarzeniach trafiają do strategii. Ma to na celu przyspieszenie reakcji strategii w porównaniu z wiadomością z giełdy.

Konceptualny model procesu realizacji transakcji wraz z umiejscowionym *OMS* przedstawia rysunek 8.



Rys. 8. Model działania komponentu Order Management System
Źródło: Opracowanie własne na podstawie [1]

Po lewej stronie na rysunku 8 widoczny jest *CEP Engine*, z którego wychodzą strumienie zdarzeń pochodnych (szczegóły zlecenia handlu). Trafiają one do *OMS*, gdzie w pierwszej kolejności są sprawdzane pod względem poprawności. Kontynuacją tego procesu jest sformułowanie zlecenia i wysłanie go przez bramkę (z ang. *gateway*) do jednego z odbiorców na przykład giełdy. Komunikacja między bramką a odbiorcą odbywa się poprzez wykorzystanie protokołu *FIX*. Jako rezultat omawiany komponent otrzymuje potwierdzenia statusu realizacji zlecenia tą samą drogą i może go przekierować do interfejsu użytkownika.

¹⁶ Donadio S., Ghosh S., *Learn Algorithmic Trading*, Pact, Birmingham 2019, s. 231

3. Wybrane strategie handlu algorytmicznego

Strategie w handlu algorytmicznym należy postrzegać jako implementacje pomysłów uczestników rynku na handel. Są to predefiniowane reguły, według których silniki systemów przetwarzania zdarzeń generują sygnały handlu. Z tego względu stanowią one złożony przedmiot rozważań, ponieważ istnieje duża liczba strategii klasyfikowanych w subiektywny sposób.

Niniejszy rozdział ma na celu zaznajomienie czytelnika z wybranymi strategiami handlu algorytmicznego, które są wdrażane w *CEP Engine*.

Realizację tych zamierzeń stanowią opisy trzech wybranych grup strategii wraz z przykładami. Należą do nich: *Trend Following*, *Mean Reversion* i *Statistical Arbitrage*. Są one opatrzone własnoręcznie zaprogramowanymi wizualizacjami przy wykorzystaniu języka *Python* w *Jupyter Notebook*. W tym celu wykorzystywane są dzienne dane historyczne pochodzące ze strony *Yahoo Finance*. Obejmują one okres jednego roku począwszy od 1 stycznia 2020 do 31 grudnia 2020. Wybór tego przedziału czasu podyktowany jest pragnieniem przedstawienia czytelnym wizualizacji.

Do pozyskanych danych należą tak zwane zasadniczo identyczne akcje (z ang. *substantially identical security*) Indyjskiej firmy informatycznej *Infosys*. Te papiery wartościowe są notowane na *The New York Stock Exchange (NYSE)* oraz *National Stock Exchange of India (NSE)* pod symbolami *INFY* i *INFY.NS*. Ze względu na to, że zasadniczo są to identyczne akcje można nimi obracać na różnych giełdach mimo posiadania odmiennych symboli.

Poza notowaniami papierów wartościowych firmy *Infosys* pobierane są dane pary walutowej *USD/INR* notowanej na *NYSE* pod nazwą *USDINR=X*.

Motywacją stojącą za wyborem tych danych jest możliwość zwizualizowania działania strategii *Statistical Arbitrage* o nazwie *Cross Market Arbitrage*.

W tej części projektu dane pozyskuje się z wykorzystaniem pakietu *yahoo_fin* dedykowanemu *Python 3*. Opiera się on na technologii *API* (z ang. *Application Programming Interface*) do pobierania między innymi historycznych danych finansowych ze strony *Yahoo Finance*.

Technologię *API* opisuje strona [19]¹⁷. Według niej *API* to zestaw zdefiniowanych reguł, które wyjaśniają w jaki sposób komputery lub aplikacje komunikują się ze sobą. Interfejsy *API* znajdują się między aplikacją a serwerem sieciowym, działając jako warstwa pośrednicząca, która przetwarza transfer danych między systemami.

Aplikacja kliencka inicjuje wywołanie interfejsu *API* w celu pobrania informacji. To żądanie jest przetwarzane z aplikacji na serwer sieciowy za pośrednictwem jednolitego identyfikatora zasobów (*URI*) interfejsu *API*. Zawiera on czasownik żądania, nagłówki, a czasami treść żądania. Po otrzymaniu poprawnego żądania *API* wykonuje wywołanie do zewnętrznego programu lub serwera WWW. Serwer wysyła odpowiedź do *API* z żądanymi informacjami. *API* przesyła dane do aplikacji, która wysłała żądanie.

3.1 Trend following

Przykładem definicji *Trend Following* jest definicja podana w artykule¹⁸ napisanym przez: Simona Fonga, Yain-Whara Si oraz Jackiego Tai. Uczni określa ją *Trend following* jako filozofię handlu, zgodnie z którą decyzje kupna/sprzedaży są podejmowane na podstawie zaobserwowanych trendów. Badacze opierają swój pogląd na przekonaniu, że bardziej prawdopodobna jest kontynuacja trendu (we wzroście lub spadku cen instrumentów finansowych) niż zmiana tendencji. Rozpoznanie trendu na początku handlu i trzymanie się go ma przynieść korzyści inwestycji. Odbywa się to poprzez zmierzenie jak bardzo trend rozwija się od ostatniej zmiany oraz upewnieniu się, że jego wzrost przewyższa poprzednie zmiany. Oznacza to, że jeśli kwota wzrostu lub spadku przekracza pewien poziom to generowany jest sygnał handlu.

Autorzy wspomnianej publikacji wskazują na udział człowieka w tym procesie przejawiający się ustawianiem parametrów strategii. Jednocześnie przestrzegają przed nierozważnymi decyzjami, ponieważ zbyt rzadkie lub częste sygnały nie przynoszą zysków.

¹⁷ Application Programming Interface (API), 2020, <https://www.ibm.com/cloud/learn/api>, (dostęp 31.05.2022)

¹⁸ Fong S., Si Y., Tai J., Trend Following Algorithms in Automated Derivatives Market Trading, Expert Systems with Applications, 2012, Volume 39, s. 11378

3.1.1 Exponentially Weighted Moving Average Crossover

Według strony [23]¹⁹ *EWMA* (z ang. *Exponentially Weighted Moving Average Crossover*) jest miarą ilościową lub statystyczną stosowaną do modelowania i opisywania szeregów czasowych. Stanowi przykład funkcji rekurencyjnej, ponieważ wartość jest obliczana na podstawie poprzedniej wartości. *EWMA* zawdzięcza swoją nazwę stosowanym malejącym wykładniczo wagom. Oznacza to, że wraz ze starzeniem się punktów danych spadają ich wagi. Reprezentuje je wartość alfa (podana w równaniu (1)) regulowana przez handlarza. Podana powyżej strona wskazuje, że im mniejszy jest obowiązujący przedział czasowy tym wartość alfa jest wyższa.

Oprócz tego wspomniane źródło przedstawia przykład zastosowania *EWMA* w strategii *Crossover*. Strona obrazuje dwie przecinające się linie pokazujące, jako momenty generowania sygnałów.

$$\alpha = \frac{2}{(n + 1)} \quad (1)$$

$$EWMA_t = \alpha * r_t + (1 - \alpha) * EWMA_{t-1} \quad (2)$$

gdzie:

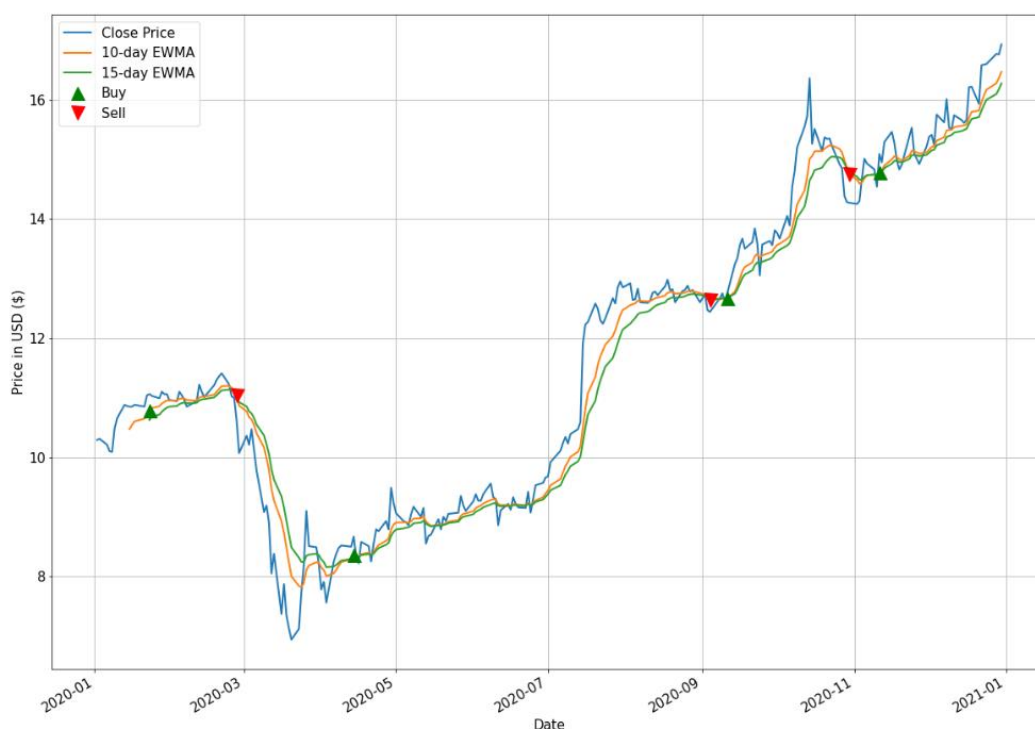
n – liczba rozważanych dni,

α – waga,

r_t – wartość szeregu czasowego w okresie t .

W tej części projektu wykorzystywane są ruchome okna o długości 10 i 15 dni oraz cena zamknięcia akcji *INFY*. Rezultaty działania algorytmu z tymi parametrami prezentuje rysunek 9.

¹⁹ Exponentially Weighted Moving Average (EWMA), 2022, <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/exponentially-weighted-moving-average-ewma/>, dostęp 18.05.2022



Rys. 9. Przykład działania EWMA Crossover na cenie zamknięcia akcji firmy Infosys (INFY)
Źródło: Opracowanie własne

Pokazana na rysunku 9 wizualizacja zmian cen akcji firmy *Infosys (INFY)* ukazuje zmieniające się notowania wraz z naniesionymi *EWMA*. W miejscach, gdzie się przecinają program generuje sygnały handlu (zielone trójkąty reprezentują kupno, a czerwone sprzedaż).

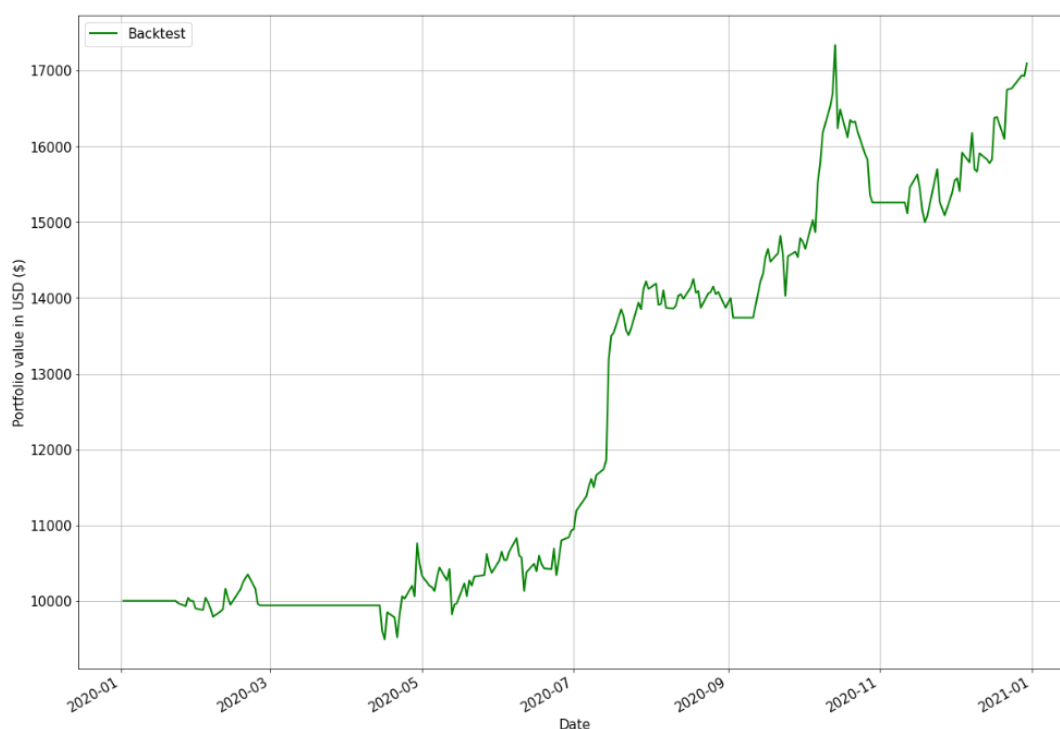
Analizując wykres można zauważyć, że zakup akcji następuje na początku trendu wzrostowego (tak zwana pozycja długa). Odwrotną sytuacją jest sprzedaż na początku opadającego trendu (tak zwana pozycja krótka). Ma to na celu przetrzymanie akcji gdy ich wartość rośnie, a następnie pozbycie się ich gdy ta wartość ma się znacząco zmniejszyć.

Istnieje grono metod historycznego testowania działania strategii (z ang. *Backtesting*). Do najpopularniejszych z nich należą skumulowane zwroty. W społeczności handlu algorytmicznego są one przedstawiane na przykładzie pojedynczej akcji lub wartości portfolio z uwzględnieniem momentów zakupu oraz sprzedaży.

Ze względu na występującą w projekcie strategię *Statistical Arbitrage*, prezentowany jest drugi sposób wizualizacji. W tym celu wykorzystywane są przykładowe parametry takie jak kapitał początkowy w wysokości 10000 dolarów oraz liczba obracanych akcji firmy *Infosys (INFY)* równa 1000. W projekcie przyjmuje się, że przy pierwszym i każdym kolejnym zakupie, 1000 akcji jest nabywanych. Analogicznie

podczas sprzedaży wszystkie akcje są zbywane, w celu uniknięcia zasygnalizowanego możliwego spadku ich wartości.

Wyniki działania algorytmu testującego strategię *EWMA Crossover*, bazującego na skumulowanych zwrotach, zaprezentowane zostały na rysunku 10.



Rys. 10. Skumulowane zwroty z akcji firmy Infosys (INFY) na podstawie strategii EWMA Crossover
Źródło: Opracowanie własne

Pokazany na rysunku 10 wykres pozwala zauważyć przeważające skumulowane zwroty o wartości powyżej 10000 dolarów. Oznacza to, że program ma tak dobrane parametry, że handluje w sprzyjających okolicznościach i jest pasywny gdy pojawia się negatywny trend (poziome odcinki). Skutkuje to absolutnym zwrotem w wysokości 71,00% na koniec rozważanego okresu.

3.2 Mean Reversion

Według strony [22]²⁰ *Mean Reversion* jest teorią handlu znajdującą zastosowanie w strategiach handlu algorytmicznego. Wspomniana koncepcja głosi, że poziom cen aktywów odbiegający daleko od długoterminowej normy lub trendu, powraca ponownie do

²⁰ Chen J., Mean Reversion, 2021, <https://www.investopedia.com/terms/m/meanreversion.asp>, dostęp 21.05.2022

swojego przeciętnego stanu. Z tego względu często wykorzystuje się ją w implementacji strategii, ukierunkowanych na kupowanie aktywów tanio i sprzedawaniu ich drogo.

3.2.1 Bollinger Band

Bollinger Band szczegółowo opisuje książka²¹ autorstwa Jirika Pika i Sourava Ghosha. Według badaczy jest to jedna ze strategii typu *Mean Reversion*. Identyfikuje ona okresy krótkotrwałej zmienności. Do tego celu wykorzystywane są trzy linie: *upper band*, *middle band* i *lower band*. Środkowa linia obliczana jest jako zwykła średnia krocząca, podczas gdy pozostałe linie są jej podwojonymi odchyleniami standardowymi.

Autorzy wskazują również na sposób generowania sygnałów handlu. W momencie przekroczenia górnej granicy przez cenę danego instrumentu finansowego tworzone są sygnały sprzedaży. Dzieje się tak, ponieważ opisana sytuacja stanowi dogodny moment zbycia aktywa o podwyższonej wartości przed jej ponownym spadkiem.

Po tym zdarzeniu, gdy cena aktywa spada poniżej dolnej granicy, generowane są sygnały kupna. Wówczas możliwe jest nabycie instrumentu finansowego przed ponownym wzrostem jego wartości.

$$\text{Upper Bollinger Band} = \text{Moving Average}(\text{Close Price}, n) + m * \delta(\text{Close Price}, n) \quad (3)$$

$$\text{Middle Bollinger Band} = \text{Moving Average}(\text{Close Price}, n) \quad (4)$$

$$\text{Lower Bollinger Band} = \text{Moving Average}(\text{Close Price}, n) - m * \delta(\text{Close Price}, n) \quad (5)$$

gdzie:

n – liczba rozważanych dni,

m – wielokrotność odchyłeń standardowych,

δ – odchylenie standardowe.

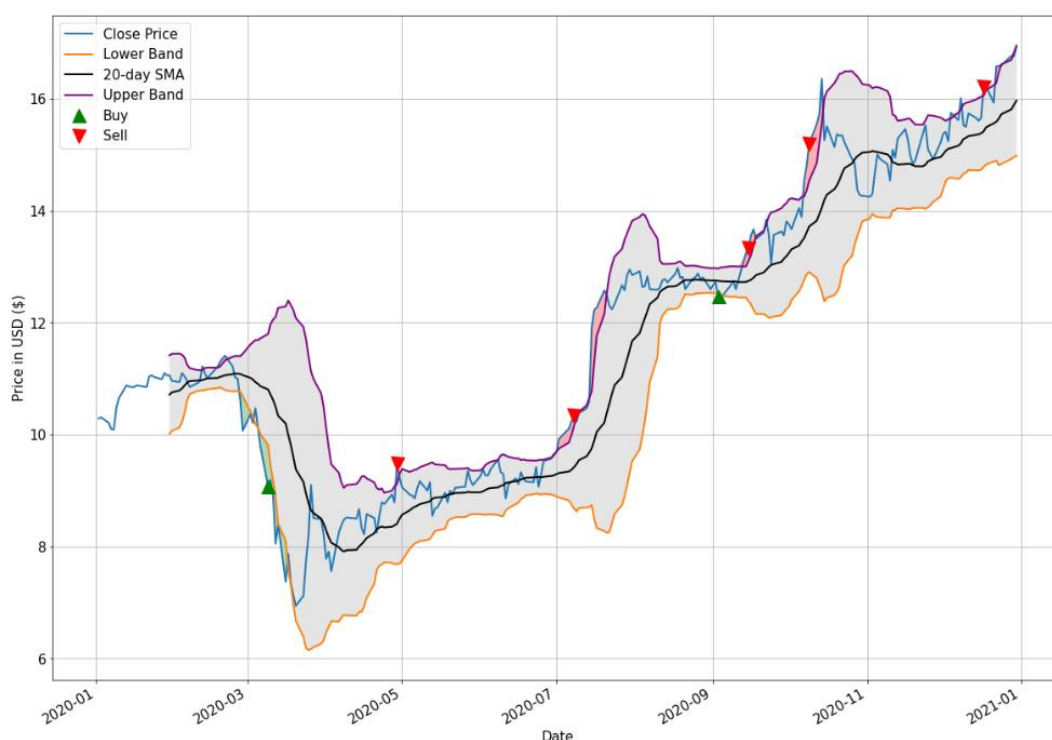
W ramach wizualizacji działania omawianej strategii wykorzystywane są wcześniej używane wielkości kapitału i akcji firmy *Infosys (INFY)*. *Bollinger Band* jest generowany na podstawie 20 dni.

Istotne dla czytelnika są momenty pojawiania się sygnałów. Na poniższym rysunku 11 kolorem zielonym i czerwonym oznaczone są obszary, gdzie to następuje.

²¹ Pik J., Ghosh S., *Hands-On Financial Trading with Python: A practical guide to using Zipline and other Python libraries for backtesting trading strategies*, Pact, Birmingham 2021, s. 292-293

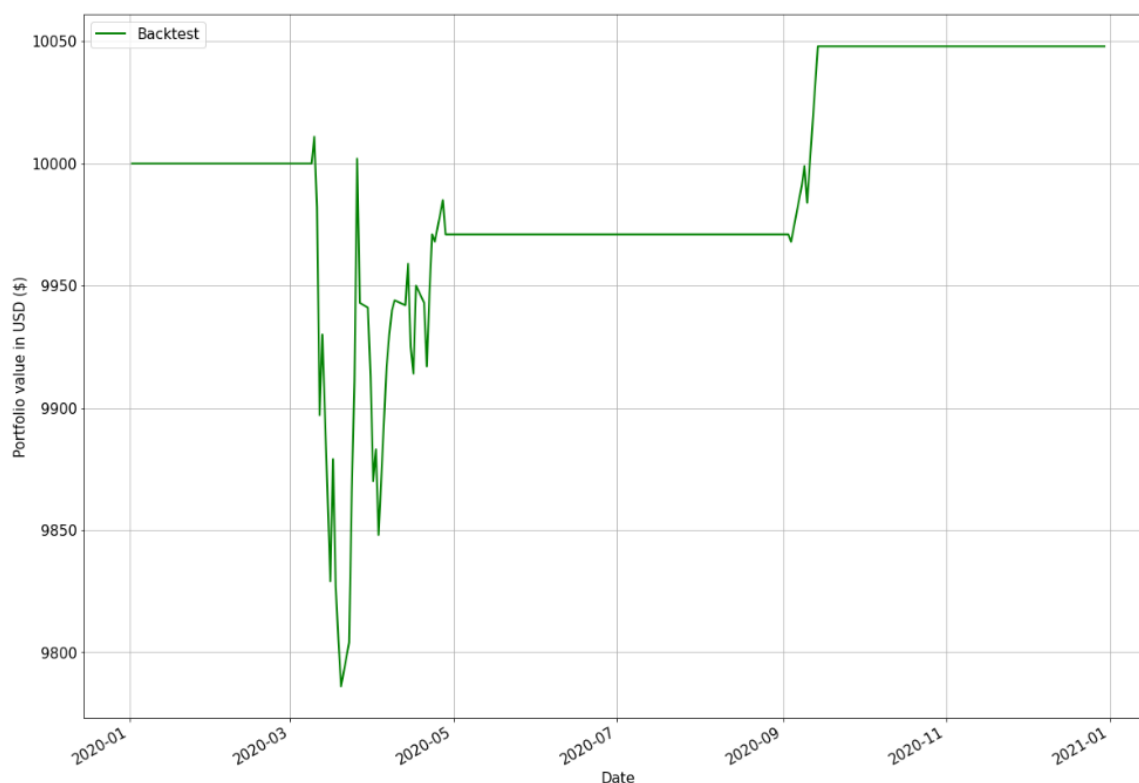
W rzeczywistości dla handlarza mało opłacalne jest dokonywanie kupna na początku spadku ceny akcji. Z tego względu popularną metodę stanowi wysyłanie zleceń po kilku powtarzających się sygnałach tego samego rodzaju.

Oprócz tego ponownie przyjmuje się, że symulacja rozpoczyna się, kiedy handlarz nie posiada wybranej liczby akcji *Infosys (INFY)*. Oznacza to, że po ich zakupieniu i sprzedaży upłynnienie tego aktywa nie jest możliwe do momentu jego ponownego nabycia. W tym projekcie faktyczny sygnał handlu generowany jest jako 6 powtórzenie tego samego sygnału zaznaczone na rysunku 11 trójkątem czerwonym lub zielonym. Efekty działania algorytmu przedstawia rysunek 11.



Rys. 11. Przykład działania Bollinger Band na cenie zamknięcia akcji firmy Infosys (INFY)
Źródło: Opracowanie własne

Na wizualizacji widoczne są między innymi sygnały kupna i sprzedaży umieszczone w czasie, kiedy można byłoby spodziewać się niewielkich zysków, a nawet strat. W celu weryfikacji wyników wykorzystywane są skumulowane zwroty zwizualizowane na poniższej grafice.



Rys. 12. Skumulowane zwroty z akcji firmy Infosys (INFY) na podstawie strategii Bollinger Band
Źródło: Opracowanie własne

Analizując wykres można zauważyć znaczący spadek wartości portfela na początku handlu. W późniejszym czasie następują odbicia, które niwelują początkowe straty a nawet przynoszą niewielkie zyski. Pod koniec rozważanego okresu absolutny zwrot wynosi 0,48% co sugeruje zmianę przyjętych parametrów strategii.

3.3 Statistical Arbitrage

Ostatnią grupą omawianych strategii jest *Statistical Arbitrage*. Wydaje się ona być najbardziej złożoną filozofią handlu algorytmicznego ze względu na rozbieżności w jej definicji oraz liczbę istniejących wariantów na przykład *Cross Market Arbitrage*.

Jedną z wielu publikacji definiujących *Statistical Arbitrage* jest artykuł²² autorstwa Gabriela Visagiego i Alwyna Hoffmana. Badacze wskazują w nim, że strategię handlu algorytmicznego należące do SA bazują na modelowaniu statystycznym. Według autorów celem tych strategii jest zapewnienie nadmiernych zwrotów z inwestycji powyżej stopy zwrotu indeksu rynkowego. Wspomniane zyski są osiągnane przy zachowaniu małej

²² Visagie G., Hoffman A., Comparison of Statistical Arbitrage in Developed and Emerging Markets, International Journal of Trade, Economics and Finance, 2017, Volume 8, s. 67

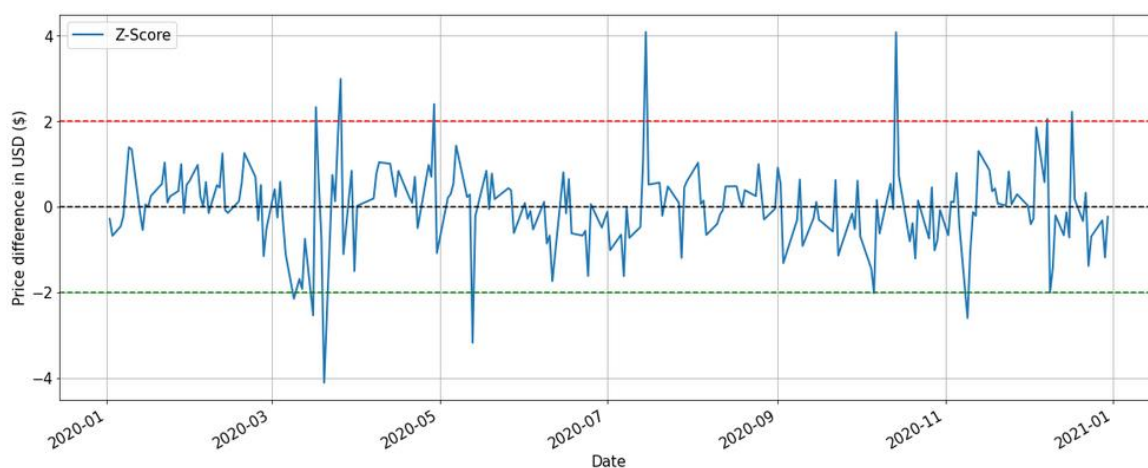
podatności portfela inwestycyjnego na zmiany rynku często w oparciu o teorię handlu *Mean Reversion*. Badacze w artykule informują, że jedną z form SA jest handel parami (z ang. *pairs trading*).

3.3.1 Cross Market Arbitrage

Koncepcję strategii *Cross Market Arbitrage* opisuje artykuł²³ autorstwa Charlesa Favreau oraz Ryana Garveya. Uczni podają, że w sytuacji gdy papiery wartościowe są kwotowane i handlowane na wielu rynkach, mogą powstać możliwości arbitrażu międzyrynkowego. Przejawia się to w przecięciu rynków, gdy najlepsza oferta cenowa papieru wartościowego na jednym rynku jest wyższa niż niższa oferta cenowa na innym rynku.

Opisaną sytuację należy interpretować jako okazję do handlu, gdzie wykorzystywane są różnice w cenach zwane *spreadem*. Dzięki niemu podmioty mogą generować zyski.

W ramach przykładu pod uwagę brane są tak zwane zasadniczo identyczne akcje firmy *Infosys* o oznaczeniach *NYSE* i *NSE*. Stanowią one silnie skorelowaną parę akcji, która w prosty sposób pozwala na wizualizację działania strategii *Cross Market Arbitrage*. Uwzględniając definicję *Statistical Arbitrage*, na podstawie *spreadu*, wyliczany jest wskaźnik statystyczny *Z-Score*. Wskaźnik ten na rysunku 13 pokazuje jak daleko jego wartość znajduje się od średniej.



Rys. 13. Z-Score cen akcji firmy Infosys na NYSE i NSE
Źródło: Opracowanie własne

²³ Favreau C., Garvey R., An Examination of Cross-Market Arbitrage, Journal of Applied Business and Economics, Volume 21, s. 30

Warto zauważyć, że akcje na *NYSE* są notowane w dolarach, a akcje na *NSE* w rupiach indyjskich. Z tego względu podczas obliczania *spreadu* do zamiany rupii na dolary wykorzystywana jest para walutowa $USDINR=X$.

Otrzymany wskaźnik *Z-Score* umożliwia zbadanie stacjonarności szeregu czasowego za pomocą testu *ADF* (z ang. *Augmented Dickey-Fuller Test*). Hipoteza zerowa tego testu dotyczy braku stacjonarności badanego szeregu czasowego. Treść hipotezy alternatywnej zawiera stwierdzenie o stacjonarności omawianego szeregu.

W przykładzie *p-value* wyszło w przybliżeniu $3,67e-14$ i było mniejsze od wartości alfa równej 0,05. Na tej podstawie odrzucono hipotezę zerową. Ustalono, że rozważany szereg czasowy *Z-Score* jest stacjonarny. Oznacza to, że para akcji jest w małym stopniu podatna na zmiany rynku.

Obserwując rysunek 13 można zauważyć zielone i czerwone linie o wartości 2,0 i 2,0. Stanowią one parametry, za pomocą których można regulować częstość i wielkość handlu. W miejscach, gdzie wskaźnik *Z-Score* przecina te linie generowane są odpowiednie sygnały. Wyniki działania algorytmu bazującego na tych sygnałach pokazane zostały na rysunku 14 zawierającym badany szereg czasowy.

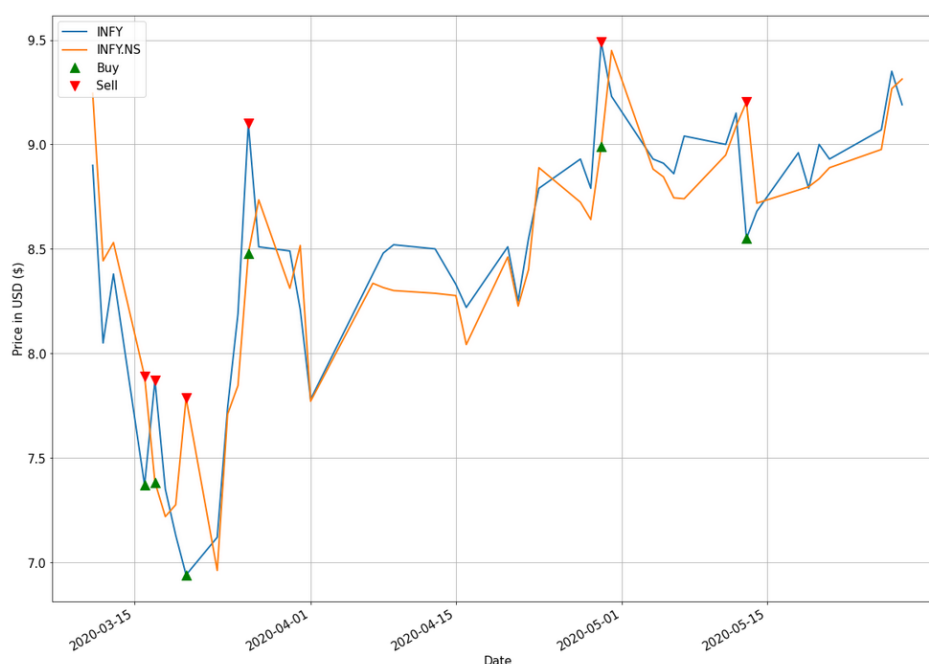


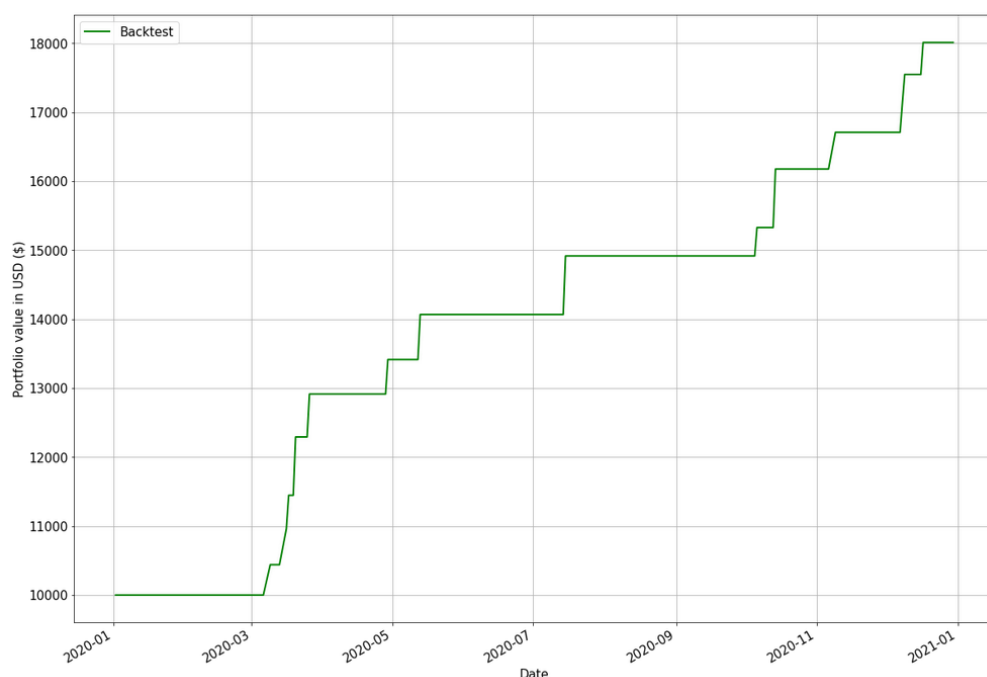
Rys. 14. Przykład działania Cross Market Arbitrage na cenach zamknięcia akcji Infosys notowanych na NYSE i NSE

Źródło: Opracowanie własne

Rysunek 14 ukazuje, że oba notowania pozostają blisko siebie. Zamieszczone sygnały wskazują momenty, w których następuje kupno tańszej akcji firmy *Infosys* na jednym rynku i ich sprzedaż po wyższej cenie na drugim rynku. Sygnał kupna może się pojawić zarówno w przypadku notowań na NYSE jak i NSE. Ta sama zasada dotyczy sygnałów sprzedaży.

Ze względu na wysoką korelację prezentowanych notowań, które na rysunku 14 w większej części się pokrywają na rysunku 15 prezentowany jest fragment tego wykresu, gdzie proces wykrywania sygnałów jest lepiej widoczny.





Rys. 16. Skumulowane zwroty z portfela akcji firmy Infosys na podstawie strategii Cross Market Arbitrage
Źródło: Opracowanie własne

Analizując wykres pokazany na rysunku 16 można zaobserwować stale rosnący trend wartości portfela. Potwierdza on panującą opinię, że ryzyko stosowania tej strategii jest bliskie 0. Ostateczne potwierdzenie skuteczności *Cross Market Arbitrage* stanowi absolutny zwrot w wysokości 80,13% na koniec rozważanego okresu.

Podsumowując ten rozdział należy pamiętać o kilku istotnych faktach. Przede wszystkim są to symulacje historyczne, które metodą prób i błędów pomagają handlarzowi ustawiać korzystne parametry strategii. Co więcej w takich testach zwykle nie są brane pod uwagę różne rodzaje kosztów, które mogą obniżać zwroty.

Wyniki strategii z obranymi parametrami prezentuje tabela 1. Zawiera ona informacje na temat wartości aktywów pod koniec okresu objętego badaniem oraz zwroty absolutne z inwestycji.

Tabela 1. Zyski i straty wynikające ze stosowania strategii z obowiązującymi parametrami dla akcji INFY

Nazwa strategii	Końcowa wartość aktywów	Zwrot absolutny
EWMA Crossover	17100,00	71,00
Bollinger Band	10048,00	0,48
Cross Market Arbitrage	18012,73	80,13

Źródło: Opracowanie własne

Zaprezentowane wyniki w szczególności zależą istotnie od obranych parametrów, czyli w momencie wybrania innych parametrów zwroty absolutne ze strategii mogłyby być inne. W tym przypadku wybrane parametry pozwoliły na wzrost kapitału od 10000 USD do wartości podanych jako końcowa wartość aktywów w tabeli 1. Wśród zaimplementowanych strategii największy zysk przyniosła strategia *Cross Market Arbitrage*, podczas gdy najslabszy wynik został uzyskany przez strategię *Bollinger Band*.

Nie ma gwarancji, że na podstawie danych historycznych inwestycje przyszłe będą równie skuteczne. Inwestycja tradingowa może przynieść duże zyski, ale też co wiedzą specjaliści, może też spowodować utratę kapitału.

4. Podsumowanie

Niniejszy projekt opisuje informatyczny proces realizacji transakcji w handlu algorytmicznym. Ze względu na brak wypracowanych standardów porządkuje on wiedzę na temat wybranej dziedziny. Przede wszystkim zostaje omówiony cel powstania tej formy handlu wraz z problematyką jej stosowania. Dzięki wymienionym działaniom czytelnik, w momencie zapoznawania się z automatycznym systemem transakcyjnym, zna jego: teoretyczne przeznaczenie, wyzwania oraz możliwe zastosowania.

Bez wątpienia stworzenie drugiego rozdziału stanowi największe wyzwanie w projekcie. Spowodowane jest to istnieniem nielicznych źródeł dotyczących architektury systemów handlu algorytmicznego. Należy przy tym zaznaczyć, że w większości przypadków są to subiektywne propozycje systemów, które nie obejmują całego zagadnienia. Niemniej jednak zauważalne są obowiązujące technologie takie jak *Complex Event Processing*. Ten fakt daje możliwość zaproponowania konceptualnego modelu systemu handlu algorytmicznego. Na bazie uniwersalnych komponentów wyjaśniane są kolejne procesy przetwarzania danych w tej formie handlu. Po przeczytaniu drugiego rozdziału czytelnik posiada wstępną wiedzę na temat systemów transakcyjnych. Stanowi ona podstawę do zgłębiania tematyki wytwarzania oprogramowania z naciskiem na transakcje giełdowe. Należy zaznaczyć, że rozdział ten stanowi podwaliny poznania procesów, które zostałyby uruchomione w momencie zaimplementowania strategii inwestycyjnych w tworzonym systemie.

Ostatnim najważniejszym osiągnięciem projektu jest prezentacja jak strategię napisane przez użytkownika łączą się z elementem decyzyjnym systemu handlu algorytmicznego. Umożliwiają to zaprogramowane przykłady z wygenerowanymi sygnałami kupna i sprzedaży. Omawiany obszar wiedzy stanowi wstęp do badań nad skutecznością strategii handlu.

Literatura

- [1] Cliff D., Trelaven P.: *Technology Trends in the Financial Markets: A 2020 Vision* 2010,
- [2] Donadio S., Ghosh S.: *Learn Algorithmic Trading*, Pact, Birmingham 2019,
- [3] Donadio S., Ghosh S.: *Learn Algorithmic Trading*, Pact, Birmingham 2019,
- [4] Etzion O., Nibblet P.: *Event Processing in Action*, 1st Edition, Manning Publications Co., Stamford 2011,
- [5] Etzion O., Nibblet P.: *Event Processing in Action*, 1st Edition, Manning Publications Co., Stamford 2011,
- [6] Favreau C., Garvey R.: *An Examination of Cross-Market Arbitrage*, Journal of Applied Business and Economics 2019, Volume 21,
- [7] Fong S., Si Y., Tai J.: *Trend Following Algorithms in Automated Derivatives Market Trading*, Expert Systems with Applications 2012, Volume 39,
- [8] Hallé S.: *From Complex Event Processing to Simple Event Processing*, 2017,
- [9] Halls-Moore L. M.: *Successful Algorithmic Trading*, 2015,
- [10] Kim K., *Electronic and Trading Technology: the Complete Guide*, Academic Press, Cambridge 2007,
- [11] Leshik E., Cralle J.: *An Introduction to Algorithmic Trading: Basic to Advanced Strategies*, 1st Edition, Wiley 2011,
- [12] Pik J., Ghosh S.: *Hands-On Financial Trading with Python: A practical guide to using Zipline and other Python libraries for backtesting trading strategies*, Pact, Birmingham 2021, s. 292-293
- [13] Shen J.: *Nine Challenges in Modern Algorithmic Trading and Controls*, Algorithmic Trading and Controls 2021, Nr 1,
- [14] Trelaven P., Lalchand V., Galas M.: *Algorithmic Trading Review*, Communications of the ACM 2013, Volume 56,
- [15] Velu R., Hardy M., Nehren D.: *Algorithmic Trading and Quantitative Strategies*, 1st Edition, CRC Press, Abingdon 2020,
- [16] Visagie G., Hoffman A.: *Comparison of Statistical Arbitrage in Developed and Emerging Markets*, International Journal of Trade, Economics and Finance 2017, Volume 8, s. 67 - 72
- [17] Wang F., Dong K., Deng X.: *Algorithmic trading system: design and applications*, Frontiers of Computer Science in China 2009, Volume 3,

[18] Wang D., Zhou W., Ali S., Zhou P., Liu Y., Wang X.: *A Novel Complex Event Processing Engine for Intelligent Data Analysis in Integrated Information Systems*, International Journal of Distributed Sensors Network 2016,

Źródła internetowe

- [19] Application Programming Interface (API), 2020,
<https://www.ibm.com/cloud/learn/api>, dostęp 31.05.2022
- [20] Backtesting, <https://www.quantconnect.com/docs/v2/our-platform/backtesting/results>,
dostęp 31.05.2022
- [21] Chart Settings,
https://help.tradestation.com/10_00/eng/tswetrading/topics/chart_configuration.htm#drawing,
dostęp 13.06.2022
- [22] Chen J., Mean Reversion, 2021,
<https://www.investopedia.com/terms/m/meanreversion.asp>, dostęp 21.05.2022
- [23] Exponentially Weighted Moving Average (EWMA), 2022,
<https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/exponentially-weighted-moving-average-ewma/>, dostęp 18.05.2022
- [24] Hayes A., Electronic Communication Network,
<https://www.investopedia.com/terms/e/ecn.asp>, dostęp 08.06.2022
- [25] Limit number of trades per day in a backtest, 2015,
<http://www.amibroker.com/kb/category/analysis/page/2/>, dostęp 13.06.2022
- [26] What is FIX?, <https://www.fixtrading.org/what-is-fix/>, dostęp 07.06.2022

Spis rysunków

Rys. 1. Strona wyników Backtestingu platformy QuantConnect	9
Rys. 2. Ceny akcji instrumentu finansowego z naniesioną linią trendu na platformie TradeStation	10
Rys. 3. Wyniki testowania działania strategii na platformie AmiBroker	11
Rys. 4. Konceptualny model systemu handlu algorytmicznego	12
Rys. 5: Uproszczona architektura systemu przetwarzania strumieni zdarzeń	14
Rys. 6. Model działania komponentu Market Data Adapter.....	16
Rys. 7. Complex Event Processing Engine w modelu systemu handlu algorytmicznego ...	17
Rys. 8. Model działania komponentu Order Management System.....	18
Rys. 9. Przykład działania EWMA Crossover na cenie zamknięcia akcji firmy Infosys (INFY).....	22
Rys. 10. Skumulowane zwroty z akcji firmy Infosys (INFY) na podstawie strategii EWMA Crossover	23
Rys. 11. Przykład działania Bollinger Band na cenie zamknięcia akcji firmy Infosys (INFY).....	25
Rys. 12. Skumulowane zwroty z akcji firmy Infosys (INFY) na podstawie strategii Bollinger Band	26
Rys. 13. Z-Score cen akcji firmy Infosys na NYSE i NSE.....	27
Rys. 14. Przykład działania Cross Market Arbitrage na cenach zamknięcia akcji Infosys notowanych na NYSE i NSE	28
Rys. 15. Przykład działania Cross Market Arbitrage na cenach zamknięcia akcji w powiększeniu.....	29
Rys. 16. Skumulowane zwroty z portfela akcji firmy Infosys na podstawie strategii Cross Market Arbitrage.....	30

Spis tabel

Tabela 1. Zyski i straty wynikające ze stosowania strategii z obowiązującymi parametrami dla akcji INFY	30
---	----