



Developing an Algorithmic Trading Robot

Project Report: August - October, 2019

Submitted by:

Crish K.C. Chengappa

Grade XII (CBSE)

Delhi Public School Bangalore North

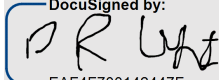
crish@intelligraft.com

On:

October 10, 2019

Approved by:

Prashant Bhat – Project Supervisor

DocuSigned by:

EAF4F700142447F... 18-Oct-2019



DEVELOPING AN ALGORITHMIC TRADING ROBOT

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.

40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



EXECUTIVE SUMMARY	3
PROJECT GOAL	3
KEY LEARNINGS	3
ABOUT INTELLIGRAFT	3
SECTION 1 – BUILDING AN AUTOMATED FX TRADING ROBOT	4
WHY FX?	4
Fig. 1 – Trading Time Zones	4
TECHNOLOGY PLATFORM.....	5
Fig. 2 – MetaTrader4 and MQL4	5
EA GENERATION WIZARD.....	6
Fig. 3 – The EA Generation Wizard.....	6
CLOUD SERVERS AND NETWORK LATENCY.....	7
Fig. 4 – Network latency with an on-premise installation	7
Fig. 5 – Network latency with a cloud-based VPS solution.....	7
SECTION 2 – TESTING THE AUTOMATED FX TRADING ROBOT.....	8
DEVELOPMENT CYCLES	8
LIVE TEST 1 - BITCOIN TRADING TESTS	8
TRADING ALGORITHM RULES.....	8
INITIAL FINDINGS.....	9
Fig 6 – BTCUSD Price History.....	9
Fig 7 – BTCUSD Trading Results Sample	9
LIVE TEST 2 - TRADING CONVENTIONAL CURRENCIES.....	10
Fig 8 – EURGBP Price History	10
Fig 9 – EURGBP Trading Results Sample.....	10
CONCLUSION	11
FUTURE DEVELOPMENTS	11
APPENDIX 1 – RESOURCES	12
APPENDIX 2 – SOURCE CODE	12

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.

40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



EXECUTIVE SUMMARY

Project Goal

This report details my experience as an intern at Intelligraft Ltd.

The goal of this project was to gain practical development experience in a software research and development lab. I was tasked with developing my own automated Expert Advisor (EA). An EA is commonly referred to in the industry as a “trading robot” and these terms are used interchangeably in this report.

I was given some initial training and supervision to achieve this task. My robot was coded in a proprietary C++ variant of the <https://www.metaquotes.net/> foreign exchange (forex) trading platform and had to make automated trading decisions in live markets, using a set of pre-defined criteria and rules.

Key Learnings

The specific learnings from this assignment can be summarised as follows:

- **Limitations of Rule-Based Algorithms** - I gained an appreciation of the limitations of rule-based systems. A strictly defined set of rules cannot self-learn, adapt and react to varying unpredictable markets, and rules become increasingly difficult to maintain as the complexity increases
- **Advantages of AI/ML Systems** – I learnt that it is ideally preferable/desirable to build AI/ML solutions which can “self-learn” how best to recognise and handle unpredictable, and varying market conditions (a topic which I intend to follow up on in future)
- **Addressing Real-World Problems** – I had to use my existing programming skills to address a real-world problem/challenge for the first time
- **Upskilling** – I had to code on a new platform (which I had not used before), teaching myself the new tools and a proprietary language, through self-study and research
- **Version Management** – I learnt how to use a cloud-based version control system and code repository
- **Cloud technology** – I experienced the benefits of cloud technology over on-premise execution
- **Basic finance trading theory** – I learnt the basics of forex trading and about currency market fluctuations

About Intelligraft

Intelligraft is our family business which I hope to run one day.

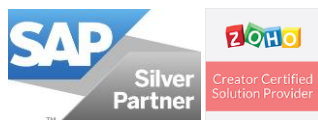
Founded in 2006 with offices in London, UK; and Bangalore, India; Intelligraft specialises in next-gen SAP, SAAS, cyber-security and C-level advisory. Intelligraft is currently investing to develop a new set of AI/ML based algorithmic forex trading technologies due for launch in 2020.

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.

40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



SECTION 1 – BUILDING AN AUTOMATED FX TRADING ROBOT

Why FX?

As an aspiring student of computer science, my goal is to specialize in AI and ML and find solutions to real world challenges.

As I take my first steps in this field, building upon the theoretical coding skills taught in school and supplemented with my own self-study, the initial challenge I faced was to find an appropriate and accessible platform which I could use to further my interest in this field.

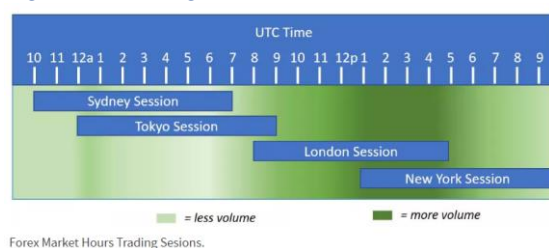
I had previously attended robotics workshops and programmed my own robots. I was extremely satisfied to use these kits to explore the basics of mechatronic programming, to program a small robot to follow a sequence of instructions to perform specific tasks, making decisions based on data it received from its sensors. However, I sought a real-world problem which could have one day be addressed with AI/ML.

Leaders in this field, such as Google and Tesla are investing billions of dollars in their AI/ML developments. A typical teenager like myself has limited access to funds and advanced technology of this nature and therefore my conclusion was that I needed to find an area where I could develop software, without major investment or barrier to entry.

The foreign exchange markets provide a perfect learning environment for the following reasons:

- **No major investment** – many brokers allow investors to trade on their FX platforms for free using trial accounts.
- **Mature and well-established tools** – the MetaTrader platform is mature and allows one to develop custom algorithms.
- **Leverage Existing Skills** – the MetaTrader MQL4 language is a derivative of C++, which I was had already been trained in.
- **Interfaces with AI libraries** – the MQL language has an interface to the Python programming language which, in turn provides a wealth of opportunity to utilise TensorFlow, a future planned development which I hope to progress to.
- **Free/Accessible training** – there are many free resources available online to upskill and explore both the business trading theories and technological aspects of this field.
- **Commercial Relevance** – FX trading is an enormous market, one which every trading nation and major financial institution has a major interest in. Banks are using increasingly sophisticated algorithmic trading systems to make intelligent decisions to trade forex in high volumes and in high frequency.
- **Real-time Live Data** – the FX markets on which I test my robots and algorithms on, are real-time live markets, not simulations.
- **Long Market Opening Hours** – the FX market is traded 24 hrs per day (except at weekends), unlike the stock markets. Markets open in Sydney at Monday 08:00 local time (UTC+10). The overlapping time-zones of other markets in later time-zones (Tokyo, Frankfurt, London and New York), allow FX to be traded 24 hrs per day, until the close of the NY market at 17:00 local time (UTC-4).

Fig. 1 – Trading Time Zones

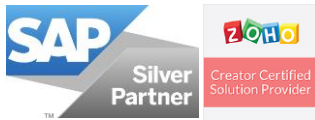


Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.

40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



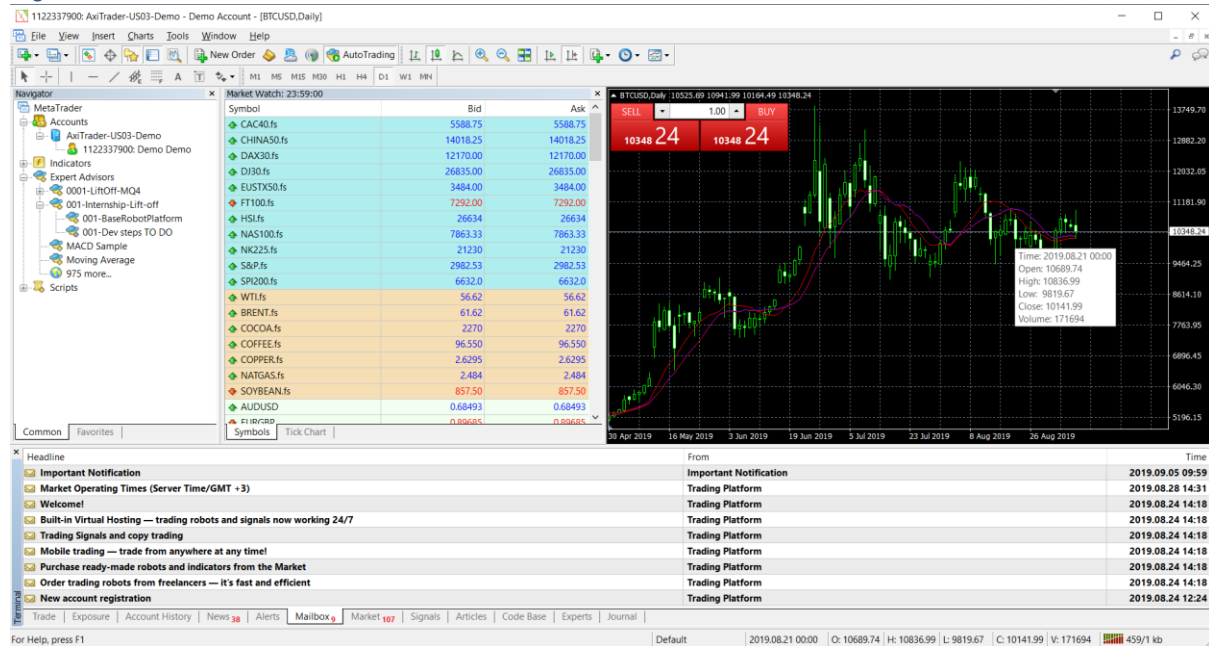
Technology Platform

The technology platform used for this assignment is MetaTrader 4 (MT4).

MT4 is a platform for trading Forex, analysing financial markets and automation using Expert Advisors.

MT4 can be programmed using a proprietary language called MQL4, which is a derivative of C++.

Fig. 2 – MetaTrader4 and MQL4



MQL4 Reference

- MQL5 features
- Language Basics
- Constants, Enumerations and Structures
- MQL4 programs
- Predefined Variables
- Common Functions
- Array Functions
- Conversion Functions
- Math Functions
- String Functions
- Date and Time
- Account Information
- Checkup
- Market Info
- Timeseries and Indicators Access
- Chart Operations
- Trade Functions
- Global Variables of the Terminal
- File Functions
- Custom Indicators
- Object Functions
- Technical Indicators
- Working with Events
- Obsolete functions
- List of MQL4 Functions
- List of MQL4 Constants

MQL4 Reference

MetaQuotes Language 4 (MQL4) is a built-in language for programming trading strategies. This language is developed by MetaQuotes Ltd., based on their long experience in the creation of online trading platforms. Using this language, you can create your own Expert Advisors that make trading management automations and are perfectly suitable for implementing your own trading strategies. Besides, using MQL4 you can create your own technical indicators (custom indicators), scripts and libraries.

MQL4 contains a large number of functions necessary for analyzing current and previously received quotes, and has built-in basic indicators and functions for managing trade orders and controlling them. The MetaEditor (text editor) that highlights different constructions of MQL4 language is used for writing the program code. It helps users to orientate themselves in the expert system text quite easily.

The brief guide contains functions, operations, reserved words, and other language constructions divided into categories, and allows finding the description of every used element of the language.

Programs written in MetaQuotes Language 4 have different features and purposes:

- Expert Advisor is a mechanical trading system linked up to a certain chart. An Expert Advisor starts to run when an event happens that can be handled by it: events of initialization and deinitialization, event of a new tick receipt, a timer event, depth of market changing event, chart event and custom events.
- An Expert Advisor can both inform you about a possibility to trade and automatically trade on an account sending orders directly to a trade server. Expert Advisors are stored in `terminal_directory\MQL4\Experts`.
- Custom Indicator is a technical indicator written independently in addition to those already integrated into the client terminal. Like built-in indicators, they cannot trade automatically and are intended for implementing of analytical functions only.
- Custom Indicators are stored in `terminal_directory\MQL4\Indicators`.
- Script is a program intended for a single execution of some actions. Unlike Expert Advisors, scripts do not process any actions, except for the start event (this requires the OnStart handler function in a script). Scripts are stored in `terminal_directory\MQL4\Scripts`.
- Library is a set of custom functions intended for storing and distributing frequently used blocks of custom programs. Libraries cannot start executing by themselves.
- Libraries are stored in `terminal_directory\MQL4\Libraries`.
- Include file is a source text of the most frequently used blocks of custom programs. Such files can be included into the source texts of Expert Advisors, scripts, custom indicators, and libraries at the compiling stage. The use of include files is more preferable than the use of libraries because of additional burden occurring at calling library functions.
- Include files can be stored in the same directory as a source file - in this case the `#include` directive with double quotes is used. Another place to store include files is

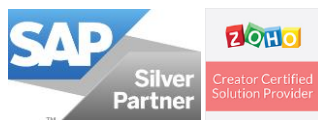
Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.

40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE

Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



EA Generation Wizard

As a developer, one can write Expert Advisors (EAs) which are commonly known as “trading robots”.

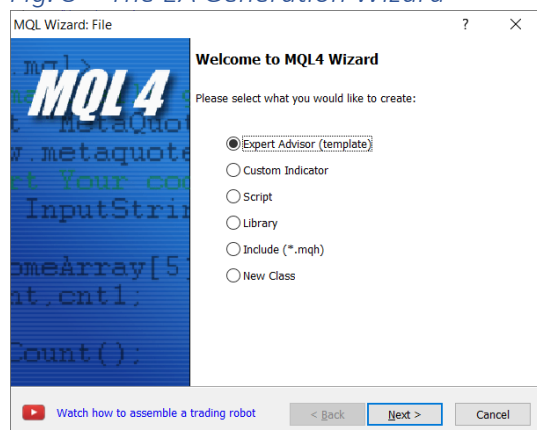
An Expert Advisor is attached to a specific trading chart which shows the real-time price movements of a specific currency pair.

I used the EA generation wizard to create a template EA which is shown in [Fig 3](#).

The blank template provides an event-driven framework into which my algorithm was programmed. It took a few weeks of learning to develop this robot and it involved a lot of testing, debugging and refinement of the code – a great learning experience.

The complete source code for my live robot is included in [Appendix 2 – Source Code](#).

Fig. 3 – The EA Generation Wizard



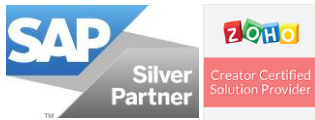
```

1 //+-----+
2 //|          Blank EA.mq4 |
3 //|          Copyright 2019, MetaQuotes Software Corp. |
4 //|          https://www.mql5.com |
5 //+-----+
6 #property copyright "Copyright 2019, MetaQuotes Software Corp."
7 #property link      "https://www.mql5.com"
8 #property version   "1.00"
9 #property strict
10 //+-----+
11 //| Expert initialization function |
12 //+-----+
13 int OnInit()
14 {
15     //--- create timer
16     EventSetTimer(60);
17
18     //---
19     return(INIT_SUCCEEDED);
20 }
21 //+-----+
22 //| Expert deinitialization function |
23 //+-----+
24 void OnDeinit(const int reason)
25 {
26     //--- destroy timer
27     EventKillTimer();
28 }
29 //+-----+
30 //| Expert tick function |
31 //+-----+
32 void OnTick()
33 {
34     //---
35 }
36 //+-----+
37 //| Timer function |
38 //+-----+
39 void OnTimer()
40 {
41     //---
42 }
43 //+-----+
44 
```

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



Cloud Servers and Network Latency

To ensure that a deployed EA continues to run without interruption, with minimal network latency, it is advisable to run it on a specialised cloud server, known as a Virtual Private Server (VPS). This can be rented relatively cheaply.

Initial trials were performed using my laptop (on-premise) and also an Amazon Web Services (AWS) free tier server. However, it became apparent that the network response times were slow (sometimes by a few hundred milliseconds) as seen in [Fig 4.](#)

Fig. 4 – Network latency with an on-premise installation

2019.10.09 13:18:54.475	'1122374781': login on AxiTrader-US03-Demo through US Datacentre 01 (ping: 208.81 ms)
-------------------------	---

A specialised VPS solution is required for professional live trading operations and this delivered a vastly improved performance – less than 10 milliseconds.

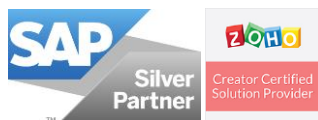
Fig. 5 – Network latency with a cloud-based VPS solution

Server	Hoster	Price, USD	Ping, ms
Netherlands	MQ5 community	10.00	0.48
Netherlands	Webzila Inc.	10.00	0.66
United Kingdom	Webzila Inc.	10.00	6.40
Frankfurt	Webzila Inc.	10.00	8.70

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



SECTION 2 – TESTING THE AUTOMATED FX TRADING ROBOT

Development Cycles

The product was developed using a sequence of iterations, with ongoing development, testing and refinement during each cycle. This resulted in the final version as shown in [Appendix 2 – Source Code](#).

With each development iteration, the EA was run and tested against live market data in real-time, to identify issues and refine the product in a series of sprints.

Once the final version was ready, it was executed against live forex markets with a demo account, with the intention of generating profitable trades.

The results are detailed below in [Live Test 1 - Bitcoin Trading Tests](#) and [Live Test 2 - Trading Conventional Currencies](#).

Live Test 1 – Bitcoin Trading Tests

For the purposes of this assignment the plan was to use the Bitcoin to US dollar currency pair (BTCUSD) due to its volatile nature.

There are many technical indicators available in the MT4 platform for analysing the financial markets, however many of them require specialist financial training to use effectively. For the purposes of this assignment, simple moving averages were selected.

Trading Algorithm Rules

The scenario was designed to:

- reflect a high-frequency trading approach where many transactions are executed per second to deliver small incremental profits.
- be rule-based and limited to buy transactions (i.e. buying long). Short-selling and hedging were beyond the scope of this assignment.

The algorithmic rules implemented can be summarised as:

1. Compare the current Ask price (i.e. buy price) of BTCUSD with the moving average of the Ask price for the previous 10 days.
2. If the Ask price was higher than the 10 day moving average, the robot would:
Buy a lot of size 0.1
Set a take profit of 75 i.e. \$0.75
Set a stop loss of 50 points i.e. \$0.50
(where 1 lot = \$10,000 and 1 point = \$0.01)
3. The robot was set to run within the following limits:
The maximum number of orders which could open at any time was limited to 5
The maximum slippage allowed was set to 10 points i.e. \$0.10
The maximum number of orders allowed during each session was limited to 1000
The maximum run-time of the robot was set to 3600 secs (1 hr)

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336

Initial Findings

The initial intention had been to trade the BTCUSD pair, however, the sharp and ongoing decline in Bitcoin price since August 2019 made it difficult to generate profit with this set of algorithmic rules (ref: [Fig 6 – BTCUSD Price History](#)).

Fig 6 – BTCUSD Price History



Extensive testing of the robot resulted in losing trades under these conditions. As short selling and hedging were beyond the scope of this assignment, correction was not possible.

[Fig 7](#) shows the results from one such trading session – significant losses were incurred in just a few minutes.

Fig 7 – BTCUSD Trading Results Sample

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P	Close Time	Price	Commissi	Taxes	Swap	Profit
104471022	2019.10.09 10:26:45	buy	0.1	btcsud	8184.78	8169.21	8185.28	2019.10.09 10:27:23	8168.15	0	0	0	-1.66
104480499	2019.10.09 12:24:44	buy	0.1	btcsud	8196.8	8180.19	8199.3	2019.10.09 12:37:29	8180.19	0	0	0	-1.66
104480730	2019.10.09 12:26:48	buy	0.1	btcsud	8198.28	8181.71	8200.78	2019.10.09 12:27:08	8181.69	0	0	0	-1.66
104481888	2019.10.09 12:34:58	buy	0.1	btcsud	8205.02	8188.47	8207.52	2019.10.09 12:35:10	8188.41	0	0	0	-1.66
104482711	2019.10.09 12:41:06	buy	0.1	btcsud	8198.8	8182.67	8201.82	2019.10.09 12:42:09	8182.21	0	0	0	-1.66
104471086	2019.10.09 10:28:30	buy	0.1	btcsud	8178.8	8163.67	8179.82	2019.10.09 10:28:32	8162.69	0	0	0	-1.61
104471755	2019.10.09 10:41:42	buy	0.1	btcsud	8189.04	8173.45	8189.54	2019.10.09 10:41:49	8172.91	0	0	0	-1.61
104481283	2019.10.09 12:30:49	buy	0.1	btcsud	8211.28	8195.43	8214.56	2019.10.09 12:30:50	8195.17	0	0	0	-1.61
104479079	2019.10.09 12:14:53	buy	0.1	btcsud	8207.56	8191.93	8208.31	2019.10.09 12:14:54	8191.69	0	0	0	-1.59
104470681	2019.10.09 10:22:19	buy	0.1	btcsud	8179.08	8163.66	8179.58	2019.10.09 10:22:28	8163.41	0	0	0	-1.57
104471840	2019.10.09 10:43:16	buy	0.1	btcsud	8194.28	8178.95	8195.04	2019.10.09 10:43:18	8178.71	0	0	0	-1.56
104471494	2019.10.09 10:35:42	buy	0.1	btcsud	8175.56	8159.93	8176.06	2019.10.09 10:37:48	8177.15	0	0	0	0.16
													-17.69

Based on these ongoing losses, a decision was taken to trade with other currency pairs.

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336

Live Test 2 - Trading Conventional Currencies

As the trading robot was designed to work on any trading asset supported by the broker, additional tests were performed on other currency pairs.

The EURGBP was selected due to the volatility caused by ongoing Brexit negotiations as seen in [Fig 8](#).

Fig 8 – EURGBP Price History



[Fig 9](#) shows a profitable trading session from trading the EURGBP currency pair, achieved in just a few minutes.

Fig 9 – EURGBP Trading Results Sample

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P	Close Time	Price	Commission	Taxes	Swap	Profit
104483045	2019.10.09 12:44:17	buy	0.1	eurgbp	0.89811	0.89751	0.89889	2019.10.09 13:12:49	0.89846	0	0	0	4.28
104483339	2019.10.09 12:46:17	buy	0.1	eurgbp	0.89805	0.89743	0.89881	2019.10.09 13:12:52	0.89848	0	0	0	5.25
104483638	2019.10.09 12:48:18	buy	0.1	eurgbp	0.89817	0.89753	0.89892	2019.10.09 13:12:53	0.89848	0	0	0	3.79
104484369	2019.10.09 12:54:18	buy	0.1	eurgbp	0.89829	0.89766	0.89903	2019.10.09 13:12:54	0.89848	0	0	0	2.32
104484613	2019.10.09 12:56:19	buy	0.1	eurgbp	0.89807	0.89744	0.89882	2019.10.09 13:12:56	0.89849	0	0	0	5.14
104484822	2019.10.09 12:58:19	buy	0.1	eurgbp	0.89802	0.8974	0.89877	2019.10.09 13:12:57	0.89848	0	0	0	5.62
104485170	2019.10.09 13:00:19	buy	0.1	eurgbp	0.8981	0.89748	0.89885	2019.10.09 13:12:58	0.89848	0	0	0	4.64
104485475	2019.10.09 13:02:22	buy	0.1	eurgbp	0.89824	0.89762	0.89899	2019.10.09 13:13:00	0.89849	0	0	0	3.06
104485673	2019.10.09 13:04:22	buy	0.1	eurgbp	0.89835	0.89773	0.89911	2019.10.09 13:13:02	0.89853	0	0	0	2.2
104485869	2019.10.09 13:06:28	buy	0.1	eurgbp	0.89847	0.89786	0.89921	2019.10.09 13:13:06	0.89857	0	0	0	1.22
104486070	2019.10.09 13:08:29	buy	0.1	eurgbp	0.89861	0.89799	0.89937	2019.10.09 13:13:11	0.89861	0	0	0	0
													37.52

The EURGBP tests provided satisfactory results, proving that it was possible to generate profit with this algorithm in a rising market.

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



CONCLUSION

The conclusions of this project are:

- Algorithmic FX trading is interesting and fun and a great way to develop one's coding and automation skills in a real-life scenario.
- This is a new field which is still evolving and thus provides a wealth of opportunities to explore and develop new AI/ML trading algorithms for this industry.
- With free resources, there is a low barrier to entry, to get started.
- Market factors heavily influence whether a specific trading strategy can succeed or fail – without using short-selling and hedging, it was impossible to make reliable profits in a falling market as demonstrated by the Bitcoin trading examples.
- Market prices are heavily influenced by political events (e.g. Brexit) and sometimes by unknown factors (e.g. Bitcoin price falls)

Future Developments

The current product uses a basic algorithm. Future planned enhancements include:

- Short selling functionality
- Hedging functionality
- Use of enhanced technical indicators to determine whether to buy long, sell short or hedge
- More sophisticated decision-making functions to evaluate market trends and trading conditions, ideally utilising AI/ML

With the advent of increasingly sophisticated AI/ML systems, it is my long term goal to build on this experience and build an intelligent trading robot which can react to varying market conditions, placing trades at optimal times and also using hedging techniques to counter sub-optimal trades (i.e. losses).

This internship has empowered me with the skills to continue development in this area.

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



APPENDIX 1 – RESOURCES

MetaTrader 4	- https://www.metatrader4.com/en
MT4 Language	- https://www.mql4.com/
Trading Hours	- https://www.investopedia.com/terms/f/forex-market-trading-hours.asp
Virtual Private Cloud Servers	- https://youtu.be/yd3ar2yOpCo

APPENDIX 2 – SOURCE CODE

```
//+-----+
//|                                     0001-BaseRobotPlatform.mq4 |
//|                                     Copyright 2019, Intelligraft Ltd |
//|                                     https://www.intelligraft.com |
//+-----+
#property copyright      "Copyright 2019, Intelligraft Ltd"
#property link           "https://www.intelligraft.com"
#property version        "1.00"
#property strict
#property description    "Author: Crish Chengappa"
#property description    "Title: Internship Programme 2019 - Building an Algorithmic FX Trading Robot"

//+-----+
//| Code Conventions
//+-----+
// Prefix igv_ = global input variable parameter/argument
// Prefix egv_ = global extern variable parameter/argument
// Prefix gc_  = global constant
// Prefix gv_  = global variable
// Prefix iv_  = local variable imported as function parameter
// No prefix   = local variable

//+-----+
//| Trading Conventions
//+-----+
// 1 Point      = $0.01
// 1 Pip        = $0.0001
//+-----+

//+-----+
//--- input parameters
//+-----+
input double   igv_TakeProfitPoints = 75;
input double   igv_StopLossPoints  = 50;
input int      igv_SlippagePoints   = 10;
input int      igv_maxOpenTrades    = 5;
extern int     egv_maxTotalOrders    = 1000;
extern double   egv_LotSize          = 0.1;
extern double   egv_dispDelaySecs    = 60;
extern double   egv_maxRuntimeInSecs = 3600;
extern double   egv_orderDelaySecs   = 120;
extern int      egv_movingAvePeriod  = 10;

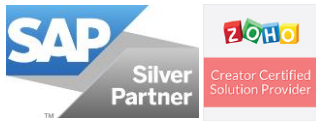
//+-----+
// Global Initialisation
//+-----+
//---Constants
const datetime gc_startTime      = TimeCurrent();
const double   gc_OpeningBal     = AccountBalance();
const string   gc_commentSpacer = "//+-----+";

//---Variables
int            gv_OpenTickets     = 0;
double         gv_nextOrderTime   = GetTickCount(); //No delay for first order
double         gv_nextDispTime    = GetTickCount()+MathAbs(egv_dispDelaySecs*1000);
double         gv_stopTime        = GetTickCount()+MathAbs(egv_maxRuntimeInSecs*1000);
string         gv_outputStr       = "";
```

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



```

string      gv_outputReport[] = {" "};
int         gv_report_length  = 0;
int         gv_BUY_count      = 0;
int         gv_FAIL_count     = 0;
double      gv_timeNow        = GetTickCount(); //Timecheck - the number of milliseconds that elapsed since the
system start

//+-----+
//| Expert initialization function
//+-----+
int OnInit()
{
    gv_outputReport[0]=gc_commentSpacer;
    gv_outputStr="Opening Balance = $" + string(gc_OpeningBal);
    gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);
    Print(formatOutputString(gv_outputStr));
    return (INIT_SUCCEEDED);
}

//+-----+
//| Expert deinitialization function
//+-----+
void OnDeinit(const int reason)
{
    printSummaryReport();
}

//+-----+
//| Expert tick function
//+-----+
void OnTick()
{
    if(buySignal()==true) // Check buy conditions
    {
        place_trade_BUY(); //--- Open a long trade when conditions are met
    }
    else
    {
        log_update();
    }
    autoStopCheck();
}

//+-----+
//| Buy Signal
//+-----+
bool buySignal()
{
    bool moreTickets      = false;
    bool AskGTMAprice     = false;
    bool nextOrderTimeReady = false;
    int periodShift       = PERIOD_CURRENT;

    if(gv_OpenTickets< igv_maxOpentrades)
        moreTickets = true;

    double getMA=iMA(_Symbol,PERIOD_CURRENT,egv_movingAvePeriod,periodShift,MODE_SMA,PRICE_OPEN,periodShift);
    if(Ask>getMA)
        AskGTMAprice = true;

    gv_timeNow=GetTickCount(); //Timecheck - the number of milliseconds that elapsed since the system start

    if(gv_nextOrderTime <= gv_timeNow)
    {
        nextOrderTimeReady = true;
    }
    return(moreTickets && AskGTMAprice && nextOrderTimeReady); //AND all results
}

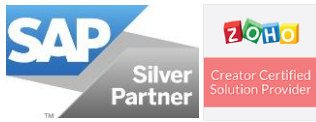
//+-----+
//| Execute Buy
//+-----+
void place_trade_BUY()

```

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
 40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
 Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



```

{
double TP_Price = Ask+(igv_TakeProfitPoints*Point);
double SL_Price = Bid-(igv_StopLossPoints*Point);

int ticket=OrderSend(Symbol(),OP_BUY,egv_LotSize,Ask,igv_SlippagePoints, SL_Price,TP_Price,gv_outputStr);
if(ticket>0) // trade is successful
{
gv_outputStr = " | BUY      order: ticket no = "+string(ticket);
gv_nextOrderTime = gv_timeNow+MathAbs(egv_orderDelaySecs*1000); //Set delay for next display
gv_BUY_count+= 1;
}
else
if(ticket<0) // trade failed
{
gv_outputStr=" | FAILED order: ticket no = " + string(ticket)+ DoubleToString(TP_Price, Digits) + "SL = "
+ DoubleToString(SL_Price, Digits);
gv_FAIL_count+=1;
}
Print(formatOutputString(gv_outputStr));
}

//+-----+
//| Log Update
//+-----+
void log_update()
{
gv_timeNow=GetTickCount(); //Timecheck - the number of milliseconds that elapsed since the system start
if(gv_nextDispTime<=gv_timeNow)
{
gv_outputStr=" | WAIT OrdersTotal() = "+string(gv_OpenTickets);
gv_nextDispTime=gv_timeNow+MathAbs(egv_dispDelaySecs*1000); //Set delay for next display
Print(formatOutputString(gv_outputStr));
}
}

//+-----+
//| Check for auto termination
//+-----+
void autoStopCheck()
{
gv_timeNow=GetTickCount(); //Timecheck - the number of milliseconds that elapsed since the system start
if(gv_timeNow>=gv_stopTime) // Runtime exceeded
{
gv_outputStr="Runtime Limited To = "+string(egv_maxRuntimeInSecs)+" (secs) - autostop invoked";
gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);
ExpertRemove();
}

if(gv_BUY_count >= egv_maxTotalOrders) // Max Orders exceeded
{
gv_outputStr="Orders Limited To = "+string(egv_maxTotalOrders)+" - autostop invoked";
gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);
ExpertRemove();
}
}

//+-----+
//| Prepare and Print Final P&L Report
//+-----+
void printSummaryReport()
{
//--- Prepare/print final report
gv_outputStr="Closing Balance = $" +string(AccountBalance());
gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);

//Profit/Loss
double profit=AccountBalance()-gc_OpeningBal;
gv_outputStr="Profit/Loss      = $" +DoubleToString(profit,2);
gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);

double profit_per_order=0;
if(gv_BUY_count!=0)
{
profit_per_order=profit/gv_BUY_count;
}
}

```

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



```
//Average Profit/Loss
gv_outputStr="Buy Orders    = "+string(gv_BUY_count)+" | Failed Orders = "+string(gv_FAIL_count)
              +" | Ave Profit per Order = "+DoubleToString(profit_per_order,2);
gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);

//Runtime & TX per sec
datetime endTime           = TimeCurrent();
datetime durationHHMMSS    = endTime - gc_startTime;
double   durationSecs      = computeTimeDiff(gc_startTime, endTime);

if(durationHHMMSS > 0)
{
    gv_outputStr="Buys/sec = " + DoubleToStr(gv_BUY_count/durationSecs,2)
                +" | Fails/sec = " + DoubleToStr(gv_FAIL_count/durationSecs,2)
                +" | Total TX/sec = " + DoubleToStr((gv_BUY_count+ gv_FAIL_count)/durationSecs,2);
    gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);
}

gv_outputStr="Start = "+TimeToStr(gc_startTime,TIME_SECONDS)+" | End = "+TimeToStr(endTime,TIME_SECONDS)+" |
duration = "+TimeToStr(durationHHMMSS, TIME_SECONDS);
gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);

// Close the report
gv_outputStr=gc_commentSpacer;
gv_report_length=ArrayAppend(gv_outputReport,gv_outputStr);

/-- Print the final report
for(int i=0; i<ArraySize(gv_outputReport); i++)
{
    int field_width=StringLen(string(gv_report_length));
    string prefix_line_no=string(i);
    do
    {
        prefix_line_no="0"+prefix_line_no;
    }
    while(StringLen(prefix_line_no)<field_width);
    Print("\n"+prefix_line_no+": "+gv_outputReport[i]);
}

//+-----+
//| string Array Dynamic Extender
//+-----+
int ArrayAppend(string &array[],string nextElement, string errMessage = "Error - failed to resize array")
{
    int newSize=ArraySize(array)+1;
    int reserveSize=100; // parameter avoids physical reallocations of memory with every resize - reserves RAM for
    faster execution

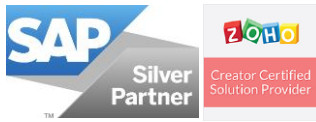
    int rc=ArrayResize(array,newSize,reserveSize);

    if(rc!=-1)
    {
        array[newSize-1]=nextElement;
    }
    else
    {
        Print(errMessage + " : "+ nextElement);
    }
    return ArraySize(array);
}
```

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
 40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
 Tel: +44 20 8133 4205

England and Wales Registration no. 5900336



```
//+-----+
//| Format Output String
//+-----+
string formatOutputString(string suffix="")
{
    string localOutputStr=StringConcatenate(
        " Bid = "+DoubleToStr(Bid, _Digits)+
        " Ask = "+DoubleToStr(Ask, _Digits)+
        " "+
        suffix
    );
    return (localOutputStr);
}
//+-----+

//+-----+
//| MQL4 version - Compute time difference in secs or mins
//+-----+
double computeTimeDiff(datetime timeStart, datetime timeEnd, int returtype=TIME_SECONDS)
{
    datetime    duration = timeEnd - timeStart;

    double      timeDiff = (TimeDay(duration) * 24 * 60 * 60) + (TimeHour(duration) * 60 * 60) +
(TimeMinute(duration) * 60) + TimeSeconds(duration);

    if(returtype == TIME_MINUTES)
    {
        timeDiff = NormalizeDouble(timeDiff/60,2);
    }
    return      timeDiff;
}

//+-----+
//| End of program
//+-----+
```

Project Report by: Crish KC Chengappa | E-mail: crish@intelligraft.com

Intelligraft Ltd.
40 Lower Ground Floor, Bloomsbury Way, London, England, WC1A 2SE
Tel: +44 20 8133 4205

England and Wales Registration no. 5900336