

جامعة الأخوين

٢٠٠٨٠٤٢١٣ ٥٧٠٤٠٦

**AL AKHAWAYN
UNIVERSITY**

SCHOOL OF SCIENCE AND ENGINEERING

**FORECASTING THE VOLATILITY OF STOCK PRICE
INDEX A HYBRID MODEL INTEGRATING LSTM
WITH MULTIPLE GARCH-TYPE MODELS**

Capstone Design

April 2022

S. Maizi

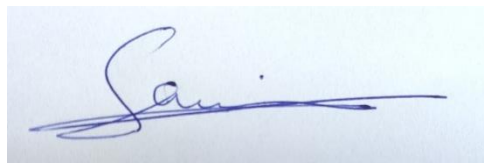
Dr. M. Azzouz, Dr. L. Laayouni

STOCK PRICE INDEX VOLATILITY FORECAST WITH LSTM AND GARCH

Capstone Report

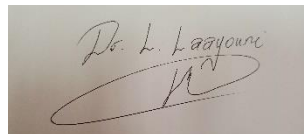
Student Statement:

“I, Sami Maizi, affirm that I have applied ethics to the design process and in the selection of the final proposed design. I also affirm that I held the safety of the public to be paramount and addressed this in the presented design wherever may be applicable.”



Sami Maizi

Approved by the Supervisor(s)



Dr. L. Laayouni



Dr. M. Azzouz

ACKNOWLEDGEMENTS

I would like to sincerely thank both my supervisors, Dr. Azzouz and Dr. Laayouni for their unconditional support during this project.

I would also like to thank, but also apologize to Dr. Alj and Dr. Lhou for their patience regarding the submission of this project and the work they've done in coordinating and being part of this work.

I would also like to thank the Al Ghurair Foundation for Education for their support during my AUI journey, especially during times that were very difficult for me. Special thanks go to Mrs. Asmae El Mahdi and Dr. Sendide for their great support throughout those years.

I also want to thank my mother for all she has done for me.

And as always, thanks be to God.

TABLE OF CONTENTS

Acknowledgements	ii
Table of Contents	iii
Abstract	v
Resume	vi
List Of Figures	vii
List Of Abbreviations And Acronyms	viii
1 Introduction	1
1.1 Background	1
1.2 Historical Setting	2
1.3 Motivation	2
2 Requirement Specifications	3
3 Feasibility Study	4
4 STEEPLE Analysis	5
4.1 Societal Implications	5
4.2 Technological Implications	5
4.3 Environmental Implications	5
4.4 Economic Implications	5
4.5 Political Implications	6
4.6 Legal and Ethical Implications	6
5 Methodology	7
5.1 Literature Review	7
5.1.1 Volatility	7
5.1.2 Stock Indexes	8
5.1.3 Hang Seng Stock Index	8
5.1.4 Financial Engineering	8
5.1.4 Deep Learning and Finance	9
5.2 Methods Used	11
5.2.1 Python	11
5.2.2 Deep Learning	12
5.2.3 Neural Networks	13

5.2.4 Regression	15
5.2.5 ARCH Model	16
5.2.6 GARCH Model	16
5.2.7 LSTM	17
5.2.8 Numpy	17
5.2.9 pandas	19
6 Predictions	19
6.1 Data Collection	21
6.2 GARCH	21
6.3 LSTM	24
7 Results	27
8 Limitations and Future Work	27
9 Conclusion	28
10 References	29
11 Appendix A: Capstone Specifications	31
12 Appendix B: Feasibility Study	32
13 Appendix C: Snapshots of the Code of the Program	33
14 Appendix D: Email Approvals	38

ABSTRACT

This document is a report of the Capstone project conducted at Al Akhawayn University in Ifrane. It is the ultimate step in order to demonstrate the technical knowledge acquired throughout the previous years studied at the University.

The project of this report consists of making predictions of the volatility of the Hong Kong stock index, the Hang Seng Index, using both GARCH models and LSTM, which is an advanced Deep Learning model compared to the state of the art model used in the financial industry, which falls within the Financial Engineering and Quantitative Finance fields.

Key Words: *GARCH, LSTM, Hang Seng, Deep Learning,*

RESUME

Ce document est un rapport du projet de fin d'étude mené à l'Université Al Akhawayn à Ifrane. C'est l'étape ultime afin de démontrer les connaissances techniques acquises tout au long des années précédentes étudiées à l'Université.

Le projet de ce rapport consiste à faire des prédictions de la volatilité de l'indice boursier de Hong Kong, l'indice Hang Seng, en utilisant à la fois les modèles GARCH et LSTM, qui est un modèle de Deep Learning avancé par rapport au modèle de pointe utilisé dans le secteur financier, ce qui relève des domaines de l'ingénierie financière et de la finance quantitative.

Mots Clés: *GARCH, LSTM, Hang Seng, Deep Learning,*

LIST OF FIGURES

Figure 2.3.3 Architecture of a Neural Network

Figure 6.2.1 Plot of the percentage change of the index

Figure 6.2.2 Plot of the partial autocorrelation

Figure 6.2.3 Plot of the prediction of the volatility using GARCH

Figure 6.2.4 Plot of the volatility of the index

LIST OF ABBREVIATIONS AND ACRONYMS

GARCH	Generalized Auto-Regressive Conditional Heteroskedasticity
ARCH	Autoregressive Conditional Heteroskedasticity
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
HSE	Hang Seng Stock Index

1 INTRODUCTION

1.1 BACKGROUND

Since the creation of financial markets, humans have searched for numerous ways to exploit them to their advantage by maximizing their profits. As the financial industry has evolved, new financial instruments and indicators have evolved in order to support financial analysts in their decision-making. One of these indicators is stock market indexes which merge multiple stock prices depending on various criteria -like the weight of each stock within the index, the valuation and size of the company as well as the industry in which it is operating- in a way that gives the financial analyst an idea about the market.

One of the core concepts in the field of finance is a risk. It is mathematically represented by the volatility of the price of any given financial instrument or asset. In other words, it represents how quickly the price changes over time. One of the motives of many financial analysts is to reduce risk as much as possible, the calculations used to determine the volatility help in the identification of the safest assets. Historically, the first way to measure the volatility was simply the historical average, which isn't very accurate. Later, the exponentially weighted average was used as it gave more importance to later values of the volatility. But the most advanced in this lineage is the GARCH model, which derived from the ARCH model

In parallel with the evolution of the financial sector, technologies have emerged at a rapid rate since humans started using computers. This has impacted thoroughly the way information is generated, stored, accessed, and exploited. As the quantity of data generated each year has increased exponentially, the field of data science has emerged in order to optimize the way this data is stored, analyzed, and exploited. Companies that use data-driven decisions have proven to be more efficient. Also, the fast evolution of modern technologies has allowed the exploitation of more data, whether in terms of volume or velocity.

One of the most advanced fields of Artificial Intelligence, which is used within data science, is Deep Learning. This subfield used neural networks which mimic the way the brain is organized in order to further optimize the learning of a certain model and give more accurate results. One of the different architectures of neural networks is the LSTM model.

The evolution of both sectors -financial and technological- gave birth to a new field that is now the standard in the financial industry: Financial engineering and quantitative finance. This field tries to make sense of financial data using very advanced technologies allied with sophisticated mathematical models. Whether it is to optimize the value of a portfolio, increase wealth, or hedge against risk, the usage of quantitative finance in the financial industry has become mandatory. A true race between multiple financial institutions has emerged in this field as the traditional methods cannot match the pace of this rapid evolution.

This project is an attempt to apply technical and analytical knowledge acquired in previous years at AUI by attempting to do a forecast of the volatility of the Hong Kong stock market index, the HSE stock market index which comprises major companies listed in Hong Kong, by using both the GARCH model -one of the most advanced financial models used to predict volatilities- and LSTM model, a deep learning algorithm. This report summarizes all the concepts and techniques used in the project, the step followed in order to achieve it, the results obtained as well as its implications from different perspectives.

1.2 HISTORICAL SETTING

The project was conducted within a time frame where the use of artificial intelligence and other advanced technologies is at its prime within the financial industry, and the competition to use the best techniques is fierce. The answer to this from the project's perspective is the use of two very advanced techniques, namely the GARCH and LSTM models.

On the other hand, the world has just witnessed the impact of the Covid 19 pandemic and is recovering after a long period and multiple waves which have influenced many economies and added more uncertainty to the financial market. By providing an elaborate way to predict volatilities, this project helps in the reduction of this uncertainty.

1.3 MOTIVATION

Besides the fact that the project uses state-of-the-art technologies, one of the main motivations behind it is that it combines two highly in-demand fields which are data science and finance. The learning process that leads to the achievement of this project, as well as all the steps that lead to it are a unique way of acquiring important skills that are very looked after in the job market today.

2 REQUIREMENT SPECIFICATIONS

When it comes to the functional requirements of the project, the project must, in the first place, provide the prediction of the volatility of the Hong Kong stock index by using the two specified techniques, namely the GARCH and the LSTM models. The program used must be free from errors, and running it over multiple times must give the same results. The program should also be able to work on multiple platforms. It should also be dynamic enough to be modified since the main purpose of the project is to help managers make decisions, and the needs can differ from time to time.

When it comes to non-functional requirements, the main output of the project to be made must be a plot that must be clear enough to read and understand the data. This means using harmonious colors, an adequate plot type, and the content of the plot should be easy to read and allow its reader to deduct information from it. The prediction data, when superposed with actual data should display a concordance that displays a certain accuracy in terms of predictions.

3 FEASIBILITY STUDY

The most basic tool that will be used for this project is my personal computer. It's a 7th gen i5 processor with 8 GB of Ram which is enough for the operations that will be conducted.

As a resource for the data, I will work with, I will rely both on YahooFinance.com and Investing.com, depending on the adequacy offered by each platform. The data will be in the .csv format, so I will have to use Excel, which is available on my computer under the university's license. I have used this tool extensively during MTH 3303, one of the uses being part of portfolio optimization in finance.

For the analysis, I will mainly use python – which I have first encountered in CSC 3323- and the variety of libraries and tools it offers like Numpy, matplotlib, pandas, Keras, and TensorFlow for Machine learning. As machine learning computations require extensive computing units, I will use Google Colab to run my program on the Cloud.

An additional resource that I will use is the book Hands-On Deep Learning for Finance by Troiano L. which contains many insights on the applications of Deep Learning in the financial sector.

As my supervisor is specialized in Mathematics, I will reach out to my supervisors if I need any help or guidance since both Finance and Deep Learning require advanced mathematical knowledge.

Overall, after assessing the resources required to undergo this project, I can conclude that it is feasible and I can move on to further steps.

4 STEEPLE ANALYSIS

4.1 Societal Implications

As working within the financial sector involves dealing with huge amounts of money that sometimes belong to many people as part of funds, it is important to be meticulous when designing the project since its usage revolves around providing accurate predictions for decision making. In the same way that it could help make better decisions, thus increasing wealth which could be used in various investments that benefit society, it could also decrease this wealth in case the project doesn't perform well. The success of this project could also incentivize other academics and professionals to pursue similar projects.

4.2 Technological Implications

Perhaps one of the main implications of this project, the project will be built using Python, which is a high-level and multipurpose programming language famous for being versatile and easy to understand thanks to its various libraries like NumPy, Pandas, matplotlib...etc. On top of that, we will use two different models, the first being GARCH which is used in the financial engineering industry for volatility predictions, in addition to LSTM, which is an advanced Deep Learning model. Cloud resources will be highly implicated during this project as the heavy computing resources required by the project can't be done on a regular machine.

4.3 Environmental Implications

Although not directly related to any environmental implication, this project, through the potential wealth it could generate, could lead to more investments in the environmental sector.

4.4 Economic Implications

One of the main economic implications of this project is the increase in efficiency of Moroccan investors by allowing them to base their decisions on advanced models, thus increasing the competitiveness of the Moroccan financial sector overall. Another implication is allowing more wealth to be generated and circulated in the economy.

4.5 Political Implications

In the same way that the success of this project could be the start of setting new regulations related to the use of technology within the financial sector, political decisions can also influence the stock market, thus influencing the results and predictions of the project.

4.6 Legal and Ethical Implications

This project is entirely made following the laws and regulations of Moroccan authorities, as the methods, data, products, and all resources used to achieve this project are all legitimate and lawful. In addition, ethics are taken into consideration in every step of the making of the project.

5 METHODOLOGY

5.1 LITERATURE REVIEW

5.1.1 Volatility

The size and frequency of price movements in an asset are both defined by volatility. If an asset has a high amount of volatility, it is regarded as riskier since its value might be distributed throughout a wide range. A low degree of volatility, on the other hand, translates to little or moderate price variations over time. When the asset's volatility is high, an option linked with the asset is more valuable because the asset holder may make a significant profit by waiting for the asset's price to spike and then purchasing it; the difference between the option's exercise price and the purchase price is maximized. A high level of volatility is regarded as unfavorable in a retirement portfolio since the portfolio's final value might be extremely difficult to forecast. To measure volatility, first, measure the difference among each data point in the sample group as well as the mean, then square the deviations to alleviate negative values, then add the squared deviations together, and ultimately divide the sum of these squared deviations by the series of data points in the sample group [1]. Knowing the way volatility functions can allow one to easily comprehend the present state of the financial sector overall, examine the risk associated with any one investment, and build an investment portfolio that would be a perfect match for growth goals and risk aversion [2]. In general, an investor has an interest in selecting the most regular investment with similar performance to avoid the issue of the entry point: this level refers to the instant the buyer purchases a security. In the case where the latter is volatile and was purchased at a high price, a portion of his bet will be lost if the volatility reduces. This danger is reduced by using a security that has a small beta [3].

5.1.2 Stock Indexes

A stock market index is a collection of equities used to gauge the performance of an industry, a stock market, or an economy. A market index is a collection of the best-performing equities on a certain exchange. A stock market index doesn't have fundamental value as an indication of many stocks. As a result, an index moves in points and represents all of its underlying assets' share values. Several indexes provide equal weight to each stock that makes up the index, while others favor firms with the highest market capitalization. Investors must use an index fund or financial derivatives such as CFDs, Futures, or ETFs to invest in a stock index. These products allow you to trade in the price fluctuations of an index without having to acquire the individual shares that make up the index. [4].

5.1.3 Hang Seng Stock Index

The Hang Seng Index is the Hong Kong Stock Exchange's major stock market index. The HSE is its identifying code, also known as RIC Reuters. As with the great majority of global indexes, the companies featured in this index are weighted by capitalisation. The HSE was founded in 1969 by Hang Seng Bank with a 100-point foundation and is made up of the largest businesses in the Chinese stock market, accounting for 65 percent of the total market capitalization. One of the largest banks in the world, Bank of China, is included in the index, as are service sector businesses like China Resources Power and industrial organizations like China Unicom, Petrochina Company Limited, COSCO Pacific Ltd, and China Coal Energy (at the time of writing). If a firm seeks finance and wants to be listed on this market, it must be among the top 90 percent of companies with the biggest capitalization and volume that have been listed for at least 24 months on the Hong Kong Stock Exchange. This index is used as an underlying asset for Asian-focused portfolios and investment products. For instance, it is the underlying asset for the HSI futures market, which is traded on the HKFE (Hong Kong Futures Exchange) and has daily trading hours of 9:45 a.m. to 12:30 p.m. and 1:00 p.m. to 4:15 p.m. Hong Kong. The HSI futures market has an average daily trading volume of 45,000 contracts and a daily volatility of 280 points (at the time of writing). It reached an all-time high of 31,958 points on October 30, 2007. [5].

5.1.4 Financial Engineering

Financial engineering is a coming together of two important concepts, finance, and engineering, that uses analytical models, economic concepts, equipment, and technical

computing solutions to address challenging and complicated financial difficulties such as irregular cash flow generation, organizing illiquid assets into liquid assets, and constructing a flawless safety net on securities [6].

5.1.4 Deep Learning and Finance

The employment of tools, even sophisticated ones, to anticipate short-term trends extremely fast runs counter to the idea of market efficiency in the trading world. Because the position formed on the basis of this forecast has an impact on the market, the forecast is swiftly absorbed into the price. As a result, generating risk-free short-term returns consistently over time is extremely tough. Machine learning's most potential applications are in the subject of portfolio management, which deals with longer-term investments. To discover the most intriguing assets, portfolio managers utilize machine learning to examine all accessible business data (financial reports, press releases, news, and even transcribed sound recordings or videos). The goal is to draw attention to the connections between a company's operational and financial history and the performance of its securities on the stock market. Expertise, gained through experience, in portfolio management, as in all domains with a high intellectual content, is primarily dependent on the rapid detection of patterns, or "patterns." We are beginning to see systems that include actual knowledge thanks to machine learning. As a result, an increasing number of asset managers are turning to machine learning, either to make or to assist investment choices, with the goal of creating algorithms that can adapt to changing environments quicker than traditional quant solutions. Recently, a hedge fund was formed in which all investment choices are done automatically by an artificial intelligence trading system. However, the fund's performance has yet to be revealed. Machine learning systems allow traders' behavior to be continually analyzed, employing not only the history of orders made but also instant messaging conversations. These solutions are far more successful than standard approaches for detecting fraudulent conduct or unapproved risk-taking, which rely on polls and a posteriori. In terms of the algorithms themselves, the most well-known problem is "overlearning" or "overfitting," in which the system gets extremely complicated and loses the ability to discern between truly relevant correlations and those that are merely "noise" in the data. However, it is mostly compliance professionals within institutions and regulators that may criticize these new tendencies. While a "traditional" program, no matter how sophisticated, can be defined in an intelligible manner and its behavior can be predicted, the same cannot always be said of machine-learning-based systems, which might appear as "black

boxes." Ironically, it will be then that human intervention will be required to explain and govern machine behavior. [12]

5.2 METHODS USED

5.2.1 Python

Guido van Rossum, a programmer, designed Python in 1991 as an open-source programming language. The show Monty Python's Flying Circus inspired the name. Because it is an interpreted programming language, it does not require compilation to run. Python code may be run on any machine using a "interpreter" application. This helps you to observe the consequences of a code modification fast. However, this makes this language slower than a compiled language such as C. Python allows programmers to concentrate on what they do rather than how they do it since it is a high-level programming language. As a result, writing programs takes less time than developing programs in another language. It is an excellent first language. Python's success stems from a number of features that assist both beginners and professionals. First and foremost, it is simple to understand and use. It has minimal capabilities, allowing you to write applications fast and with minimum effort. Furthermore, its syntax is intended to be legible and simple. Python's popularity is another benefit. This language is compatible with all major operating systems and platforms. Furthermore, while it is not the quickest language, its diversity compensates for its slowness. Finally, this language is utilized to develop professional grade software, despite the fact that it is mostly used for scripting and automation. Python is used by a vast number of developers to construct software, whether it's apps or online services. Python comes in two flavors: Python 2 and Python 3. There are several variations between these two versions. Python 2.x is the previous version of Python, which will be supported and receive official upgrades until 2020. It will most likely continue to exist informally after that day. The language's current version is Python 3.x. Many new and beneficial features are included, including improved concurrency management and a more efficient interpreter. The absence of supporting third-party libraries has long been a barrier to Python 3 adoption. Because many of them were only compatible with Python 2, the transfer was difficult. However, this issue has mostly been resolved, and there are few compelling reasons to continue using Python 2. Scripting and automation are Python's most common applications. Indeed, this language allows for the automation of online browser or application GUI interactions. Scripting and automation aren't the sole applications for this language. It's also used for developing apps, providing web services or REST APIs, and metaprogramming and code creation. Furthermore, this language is utilized in the fields of data science and machine learning. Data analysis has been one of the most common uses of

the technology as it has spread across many sectors. The great majority of data science and machine learning libraries include Python interfaces. As a result, this language has grown in popularity as the most widely used high-level command interface for Machine Learning libraries and other digital algorithms. There are several beginning books available on the internet. Finally, robotics businesses like Aldebaran employ this programming language to train their robots. This programming language was chosen by the Softbank-acquired firm to make it easier for third-party companies and amateurs to build applications. [7].

5.2.2 Deep Learning

Deep learning, often known as deep learning, is an artificial intelligence subfield (AI). This phrase refers to a set of automatic learning techniques (machine learning), which are used to model data and are based on mathematical principles. To have a deeper understanding of these strategies, we must go back to the beginnings of artificial intelligence in 1950, when Alan Turing became interested in thinking machines. Machine learning, a machine that interacts and behaves based on stored data, will be born from this contemplation. Deep learning is a sophisticated system that contains a huge network of artificial neurons and is based on the human brain. These neurons work together to absorb and memorize information, compare issues or situations to previous comparable situations, examine solutions, and solve problems as efficiently as feasible. Deep learning, like human learning, is based on lived experiences or, in the case of robots, recorded data. In the realm of information and communication technology, deep learning is extremely valuable. It's utilized in facial and speech recognition systems in certain smartphones, as well as robots, to ensure that smart equipment reacts appropriately in a given environment (for example a smart refrigerator that emits an alarm signal if it detects a door left open or an abnormal temperature within the compartments). Have you ever wondered how Facebook detects your pals in your photos? You've discovered the solution: deep learning. Deep learning is employed by researchers, particularly those who study and/or edit DNA. These technologies may also be found in automated translation systems, autos and other autonomous vehicles, medical (radio, MRI, scanner), science (searching for particles), and the arts (reproducing a work). Signals pass between neurons in the artificial brain in the same way they do in the human brain. This accomplishment is primarily due to algorithms. In the instance of visual recognition, the deep learning system must be able to recognize all existing forms from all angles in order to be effective. As a result, it will be capable of detecting a car on the road in the middle of nowhere. Only if the

machine has received significant training is this conceivable. And this entails looking at thousands of images of cars in all forms and from various perspectives. When a new image arrives, it is given to the neural network, which analyzes it and determines if the item in the centre of the frame is, in fact, a car. Is the machine a winner? She keeps her right response warm since it will come in handy when she has to recognize another automobile in the future. [8].

5.2.3 Neural Networks

The Perceptron was created in 1957. It's the first machine learning algorithm, and it was created to solve complicated pattern recognition problems. This approach will allow robots to learn to detect things in photos in the future. Unfortunately, technical resources constrained neural networks at the time. Computers, for example, lacked the processing power required to operate neural networks. This is why development in the field of neural networks has been stagnant for many years. Data Scientists didn't have the data or computational capacity to run large neural networks until the early 2010s, when Big Data and massively parallel processing became popular. During an ImageNet competition in 2012, a Neural Network beat a human for the first time in picture recognition. This is why scientists are now again concerned about this technology. Artificial neural networks are now improving and changing on a daily basis. A neural network often uses a large number of processors that run in parallel and are grouped into tiers. The first third gets unprocessed data, similar to how human optic nerves analyze visual signals. Following that, each third party receives the prior third party's information outputs. By evaluating samples to practice, the computer with the neural network learns to accomplish a task. These examples have been labeled in advance so that the network can recognize them. A neural network, for example, may be used to train a computer to recognize things. There are three unique learning strategies, though. In supervised learning, the algorithm is trained on a collection of labeled data and then modified until it can analyze the dataset and provide the intended output. The data is not labeled in the case of unsupervised learning. A cost function shows the neural network how far away it is from the desired output as it examines the data set. The network then adjusts to improve the algorithm's accuracy. Finally, the neural network is rewarded for positive outcomes and sanctioned for poor ones using the reinforced learning approach. This is what permits him to learn over time, much like a human learns from his mistakes over time. Different types of neural networks exist. The information transmission across the several layers of neurons varies depending on the kind of

network. Information goes straight from the input to the processing nodes and then to the outputs in the simplest variation, the so-called "feed-forward" neural network. Recurrent neural networks, on the other hand, preserve and feed the model the findings produced by the processing nodes. [8].

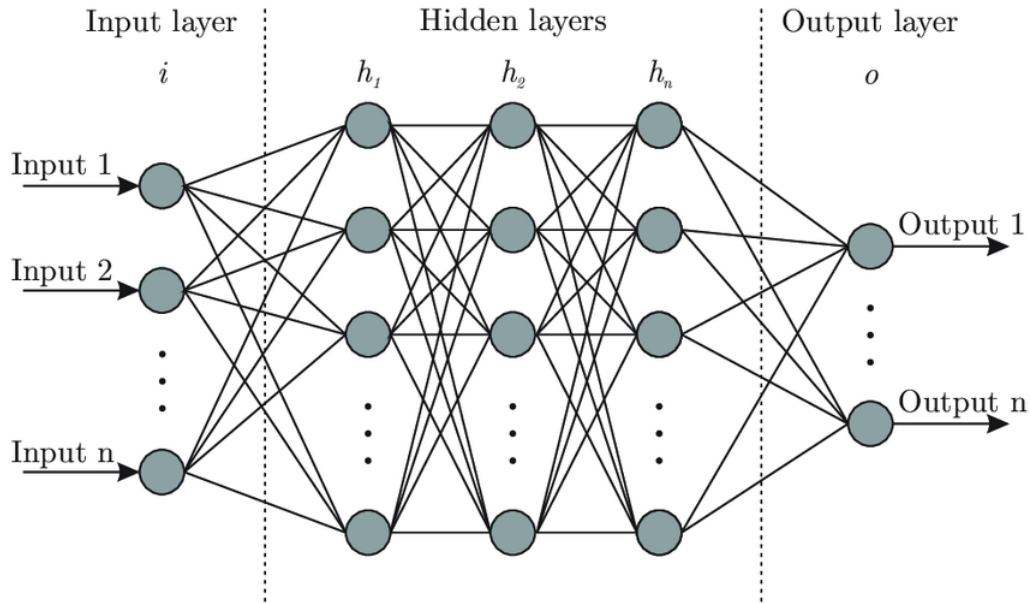


Figure 2.3.3 Architecture of a Neural Network

5.2.4 Regression

Linear regression is a model that uses current data to produce predictions or estimations. A linear link between an explained variable and an explanatory variable is constructed using a supervised learning technique. Linear regression, often known as a linear model, is a statistical model that performs prediction functions. In order to determine a trend or a predicted progression, the procedure depends on numerical numbers to produce useful estimations. The machine may then extrapolate them and predict future values using a dataset. There are various applications for linear regression. This is relevant to the development of artificial intelligence that is capable of self-learning new rules and functions. A linear model is also used in econometrics, statistics, and stock market movements. Different variables can be correlated using linear regression. The provided findings are only forecasts or approximations. As a result, despite the system's dependability, there is still some ambiguity. As a result, the linear model must be seen as a decision-making tool rather than an established reality. Simple linear regression builds its model around the use of a single explained variable (dependent) to create an independent explanatory variable, i.e. the desired prediction. The system in a multiple linear regression follows the same principles as a basic model, but it allows for the identification of at least two explanatory variables. As a result, many pieces of information based on a single element are feasible. [9].

5.2.5 ARCH Model

One of the most important benefits of a time series is that it allows us to keep track of the change of a variable over time. In economics, we immediately realized the need of constructing trustworthy variables whose evolution we can track over time (e.g., unemployment rate, inflation rate, etc.) and predict based on the values that this variable takes at different time periods. Seasonality is the most obvious example that we see in everyday life. Take, for example, the number of airline flights over the course of a year: we can see that there are times when the number of flights peaks, and these times generally correlate to vacations when people travel. When we collect data over a long period of time, we may not only find trends, but also monitor volatility and create projections for future periods. This is the important question in the estimation techniques for finance econometrics, where we frequently operate over lengthy periods of time. Time series are characterized and modelled using ARCH (AutoRegressive Conditional Heteroskedasticity) models in econometrics. These models are commonly referred to as ARCH models (Robert F. Engle, 1982), however different acronyms are used to refer to specific model architectures with a similar foundation. ARCH models are frequently used to simulate financial time series with variable volatilities, such as times of turbulence followed by periods of relative tranquility. The conditional variance at time t is variable in these models. It is dependent on the square of the process's prior successes or the square of the inventions, for example. [10].

5.2.6 GARCH Model

The GARCH model is a generalized autoregressive model that uses conditional variance to represent clusters of return volatility. To put it another way, the GARCH model determines medium-term average volatility using an autoregression based on the sum of lagged shocks and lagged variances. Heteroscedastic Conditional Heteroscedastic Generalized Autoregressive Conditional Heteroscedasticity is the English translation of Generalized Autoregressive Model. Because it considers both contemporary and past observations, it is generalized. Because the dependent variable flips on itself, the model is autoregressive. Because future variance is dependent on prior variance, it is conditional. Because the variance changes depending on the data, the model is heterocedastic. The GARCH model, as well as its extensions, are used to forecast short and medium-term volatility. Although we execute the computations in Excel, more advanced statistical tools such as R, Python, Matlab, or EViews are advised for more precise estimations. GARCH typologies are utilized based on the

variables' features. We shall employ orthogonal GARCH, for example, if we are working with interest rate bonds of various maturities. We'll utilize a different form of GARCH if we're working with actions. Financial asset returns follow a normal probability distribution with mean 0 and variance 1 and fluctuate about their mean. As a result, the returns on financial investments are utterly unpredictable. [11]. This is the formula for a GARCH model.

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha_1 \epsilon_{t-1}^2 + \dots + \alpha_q \epsilon_{t-q}^2 + \beta_1 \sigma_{t-1}^2 + \dots + \beta_p \sigma_{t-p}^2 \\ &= \omega + \sum_{i=1}^q \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^p \beta_i \sigma_{t-i}^2\end{aligned}$$

Where ω is the long run volatility, ϵ is a residual, σ_t^2 is the variance at a point in time t , α is the reaction of the volatility in regard to the unexpected return, and β is the persistence of the volatility which refers to the amount of time that the volatility could take in order to go back to the long-run volatility.

5.2.7 LSTM

When the data we are working with contains a time component, RNNs are usually used. However, the latter have restrictions in their underlying structure that render them ineffective in many scenarios. The LSTM follow with performance only equaled by their complexity. The LSTM cell is composed of a plurality of gates that contribute to the regulation of the data inflow. There are additionally two sorts of outputs (known as states): forget, input, output gate, hidden and cell state/cell. The LSTM can keep or remove information in memory using these gate actions. The information saved in the network's memory is a vector called ct : the cell's state. The network may remember data that it has previously encountered since this state is dependent on the prior state ct_1 , which is dependent on yet previous states (differently from RNNs). [12]

5.2.8 NumPy

NumPy is a widely used Python data science package. Learn all you need to know in order to master it. Data science is based on extremely difficult scientific computations. Data Scientists require strong tools to complete these computations. One such useful resource is the Python NumPy library. The acronym NumPy stands for "Numerical Python." It is a Python-based Open Source library. This tool is used to program in Python for scientific purposes,

specifically data science, engineering, mathematics, and science. This Python module comes in handy for doing mathematical and statistical computations. It's particularly useful for multiplying matrices or multidimensional arrays. It's simple to integrate with C/C++ and Fortran. This platform provides multidimensional objects in "arrays" as well as a set of Python integration tools. Simply explained, NumPy is a cross between C and Python that we utilize as a replacement for MATLAB programming. For multidimensional functions and rearrangement procedures, data in the form of numbers is handled as arrays. In the discipline of Data Science, it is an extensively utilized tool. Numpy is one of the most widely used libraries in Python. To extract significant information from data, many Data Science approaches need big tables and matrices as well as complicated calculations. NumPy makes this procedure easier by providing a wide range of mathematical functions. It is one of the most significant Python libraries for scientific computing, despite its simplicity. Furthermore, additional libraries largely rely on NumPy arrays as inputs and outputs. As a result, TensorFlow and Scikit learn to calculate matrix multiplications using NumPy arrays. NumPy also has methods that allow programmers to conduct simple and complex mathematical and statistical operations on multidimensional arrays and matrices using only a few lines of code. NumPy's major feature is the "ndarray" or "n-dimensional array" data structure. Because these tables are unique in that they are homogenous, all of the items must be of the same kind. NumPy arrays are quicker than Python lists in general. Python lists, on the other hand, are more versatile since each column may only hold data of the same kind. Here are the key features of NumPy to summarize. Ndarrays is a hybrid of C and Python that works with multidimensional and homogenous data tables (ndimensional arrays). NumPy is a Python package that allows you to conduct logical and mathematical operations on arrays and matrices. These operations are more faster and more efficient with this tool than with Python lists. NumPy arrays provide several advantages over Python lists. First and foremost, they consume less memory and storage space, which is the key benefit. A NumPy array is, in fact, smaller than a Python list. While a list may be as large as 20MB, an array is limited to 4MB. These boards are not only lightweight, but they are also simple to read and write on. Furthermore, NumPy performs better in terms of execution speed. Its use is, however, more straightforward and practical. Furthermore, it is an open source program that may be used for free. It's built on Python, a popular programming language with a plethora of high-quality libraries for any purpose. Finally, connecting existing C code to the Python interpreter is a breeze. [14].

5.2.9 pandas

Pandas is a Python computer language library devoted exclusively to data science. Learn about the purpose of this tool and why Data Scientists need it. Python is the most popular programming language for data analysis and machine learning, having been created in 1991. Several factors contribute to Data Scientists' success. First and foremost, it is a very simple language to learn. A simple and easy syntax allows even beginners to swiftly create applications. This language has a large community that has built several Data Science tools. There are, for example, data visualization tools like Seaborn and Matplotlib, as well as software libraries like Numpy. Pandas, for example, is a data manipulation and analysis package. Pandas is an open-source software library for manipulating and analyzing data in the Python programming language. It's powerful, adaptable, and simple to use. The Python language now has the ability to load, align, modify, and even combine data thanks to Pandas. When the back-end source code is written in C or Python, performance is very noteworthy. The word "Pandas" is an abbreviation of "Panel Data," which refers to datasets that contain observations from several time periods. This package was designed as a high-level tool for Python analysis. Pandas' authors hope to make it the most powerful and versatile open-source data manipulation and analysis tool available in any programming language. Pandas is commonly used for "Data Wrangling" in addition to data analysis. This phrase refers to techniques for converting unstructured data into useable information. Pandas excels in processing structured data in the form of tables, matrices, and time series in general. It works with other Python libraries as well. Pandas works with "DataFrames," which are two-dimensional tables of data with each column containing the values of a variable and each row containing a collection of values from each column. A DataFrame may hold numbers or characters as data. DataFrames are used by data scientists and programmers who are familiar with the R programming language for statistical computation to store data in grids that are simple to study. Panda is extensively used for Machine Learning because of this. This application lets you import and export data in a variety of formats, including CSV and JSON. Pandas also has Data Cleaning capabilities. This library comes in handy when working with statistical data, tabular data (such as SQL or Excel tables), time series data, and arbitrary matrix data with row and column labels. Pandas has various advantages for Data Scientists and Developers. This package makes it simple to compensate for data that is missing. It's a versatile tool since columns may be simply added or removed from DataFrames. Data label

alignment can be automated. Another benefit is a strong data aggregation tool that allows you to split, apply, and combine datasets, as well as aggregate and alter them. Differently indexed data in various Python and Numpy structures may be easily converted to DataFrame objects. Data can also be indexed or sorted using an intelligent labeling system. Datasets may be combined and restructured with ease. Data loading from CSV files, Excel files, databases, or data in HDF5 format is made easier with I/O tools. The time series functions round out the image, with features such as date range creation, frequency conversion, and statistical window displacement. Pandas is an essential Python package for Data Science because of its numerous advantages. For Data Scientists, this is a really valuable tool. [13]

6 PREDICTIONS

6.1 DATA COLLECTION

For both the GARCH and LSTM, the time series data was obtained from the website Yahoo Finance using the `datareader` function. The advantage of using this function is that it directly saves the data in the form of a pandas time-series dataframe. Also, using this method allows us to go beyond the cleaning phase since the data is accurate and there are no missing values.

```
import pandas_datareader.data as web

start = datetime(1990, 1, 1)
end = datetime(2022, 3, 10)

hse = web.DataReader('^HSI', 'yahoo', start=start, end=end)
```

6.2 GARCH

The first step was to calculate the percentage return of the HSE index and plot it

```
returns = 100 * hse.Close.pct_change().dropna()

plt.figure(figsize=(10,4))
plt.plot(returns)
plt.ylabel('Pct Return', fontsize=16)
plt.title('HSE Returns', fontsize=20)
```

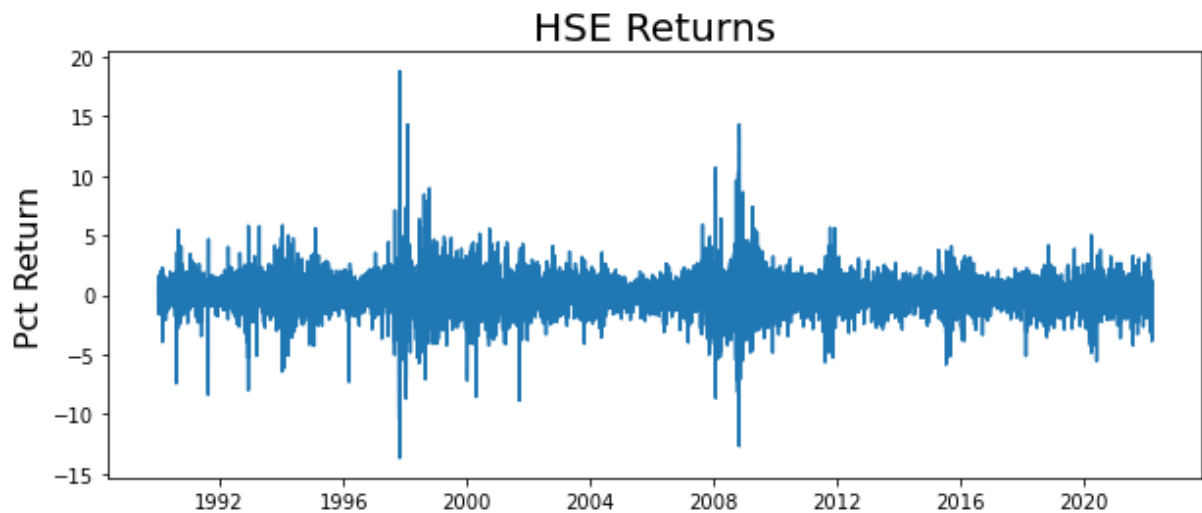


Figure 6.2.1 Plot of the percentage change of the index

In order to choose the p and q parameters of the GARCH model, we need to first plot the partial autocorrelation plot which indicates the degree of correlation between each variable and its predecessors

```
plot_pacf(returns**2)
plt.show()
```

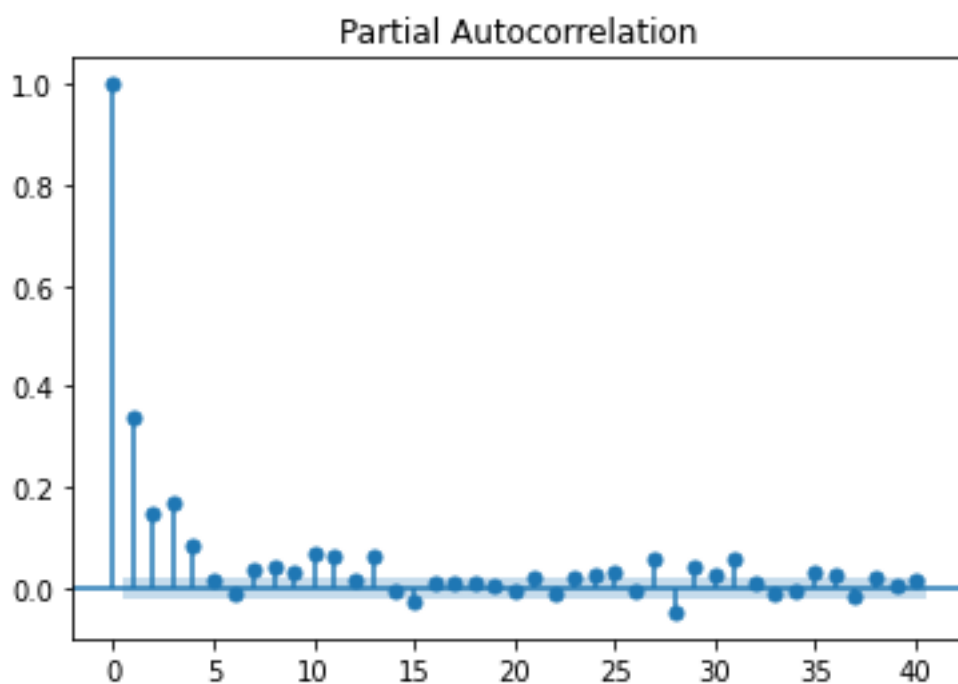


Figure 6.2.2 Plot of the partial auto-correlation

Following this autocorrelation plot, we can deduce that the best parameters are $p = 2$ and $q = 2$. Then we proceed to fitting the model and doing the forecast and plot it.

```
model = arch_model(returns, p=2, q=2)
model_fit = model.fit()
model_fit.summary()

#Rolling Forecast
rolling_predictions = []
test_size = 365*20

for i in range(test_size):
    train = returns[:-(test_size-i)]
    model = arch_model(train, p=2, q=2)
    model_fit = model.fit(dis='off')
    pred = model_fit.forecast(horizon=1)
    rolling_predictions.append(np.sqrt(pred.variance.values[-1,:][0]))

rolling_predictions = pd.Series(rolling_predictions, index=returns.index[-365*20:])

plt.figure(figsize=(10,4))
true, = plt.plot(returns[-365*20:])
preds, = plt.plot(rolling_predictions)
plt.title('Volatility Prediction - Rolling Forecast', fontsize=20)
plt.legend(['True Returns', 'Predicted Volatility'], fontsize=16)
```

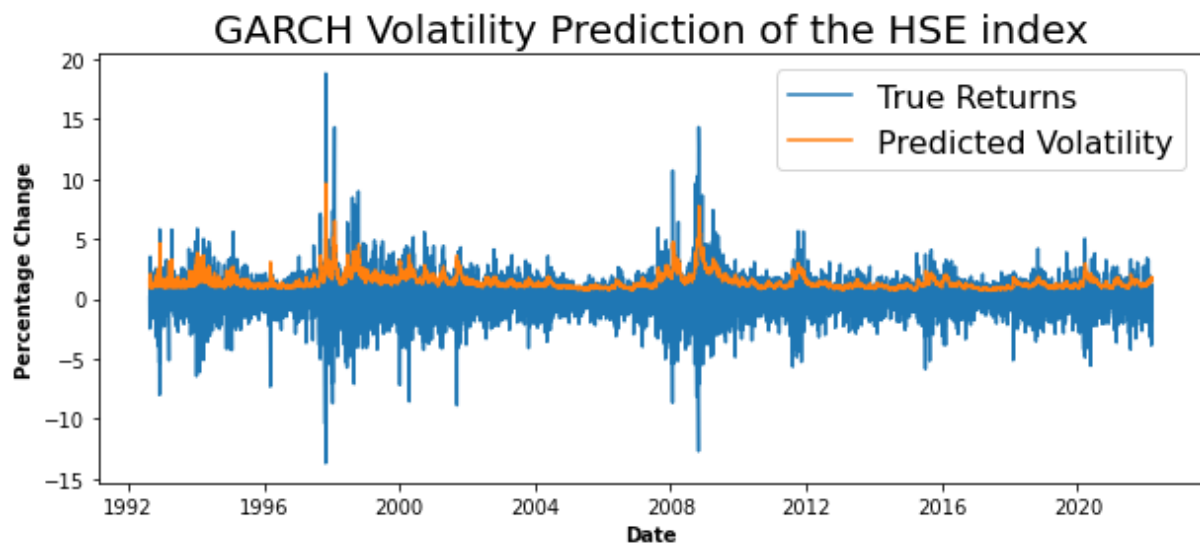


Figure 6.2.3 Plot of the prediction of the volatility using GARCH

6.3 LSTM

To work on the LSTM, we first need to extract the data of the volatility from the GARCH model using the `_volatility` attribute of the `model_fit` variable.

The first screenshot shows the Spyder Python IDE with a file named `untitled2.py` open. The code imports necessary libraries and defines a function to plot the volatility prediction. The `model_fit` variable is accessed to extract the `_volatility` attribute. The second screenshot shows the same code with the `model_fit` variable expanded in the Variable Explorer, showing the `_volatility` attribute as a Series object. The `train` variable is also expanded, showing the training data used for the model fit.

Code Snippet 1 (First Screenshot):

```
import pandas_datareader.data as web
from datetime import datetime, timedelta
import pandas as pd
import matplotlib.pyplot as plt
from arch import arch_model
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import numpy as np

plt.figure(figsize=(10,4))
true, = plt.plot(returns[-365*20:])
preds, = plt.plot(x)
plt.title('Volatility Prediction - Rolling Forecast', fontsize=16)
plt.legend(['True Returns', 'Actual Volatility'], fontsize=16)

#How to use the model
train = returns
model = arch_model(train, p=2, q=2)
model_fit = model.fit(displ='off')

pred = model_fit.forecast(horizon=7)
future_dates = [returns.index[-1] + timedelta(days=i) for i in range(7)]
pred = pd.Series(np.sqrt(pred.variance.values[-1:]), index=future_dates)

plt.figure(figsize=(10,4))
x = model_fit._volatility
x = pd.DataFrame(x)
x.set_index(model_fit._index)
print(x.info())
ind = pd.to_datetime(model_fit._index)
x.set_index(ind)
x.head()
```

Code Snippet 2 (Second Screenshot):

```
import pandas_datareader.data as web
from datetime import datetime, timedelta
import pandas as pd
import matplotlib.pyplot as plt
from arch import arch_model
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import numpy as np

plt.figure(figsize=(10,4))
true, = plt.plot(returns[-365*20:])
preds, = plt.plot(x)
plt.title('Volatility Prediction - Rolling Forecast', fontsize=16)
plt.legend(['True Returns', 'Actual Volatility'], fontsize=16)

#How to use the model
train = returns
model = arch_model(train, p=2, q=2)
model_fit = model.fit(displ='off')

pred = model_fit.forecast(horizon=7)
future_dates = [returns.index[-1] + timedelta(days=i) for i in range(7)]
pred = pd.Series(np.sqrt(pred.variance.values[-1:]), index=future_dates)

plt.figure(figsize=(10,4))
x = model_fit._volatility
x = pd.DataFrame(x)
x.set_index(model_fit._index)
print(x.info())
ind = pd.to_datetime(model_fit._index)
x.set_index(ind)
x.head()
```

Then use the extracted array and merge it with an array containing the dates of the volatility in order to generate a time series dataframe with pandas, and use it in order to generate a plot of the volatility.

```
x = model_fit._volatility
x = pd.DataFrame(x)
x = x.set_index(model_fit._index)
print(x.info())
ind = pd.to_datetime(model_fit._index)
x.set_index(ind)
plt.plot(x)
data = x
```

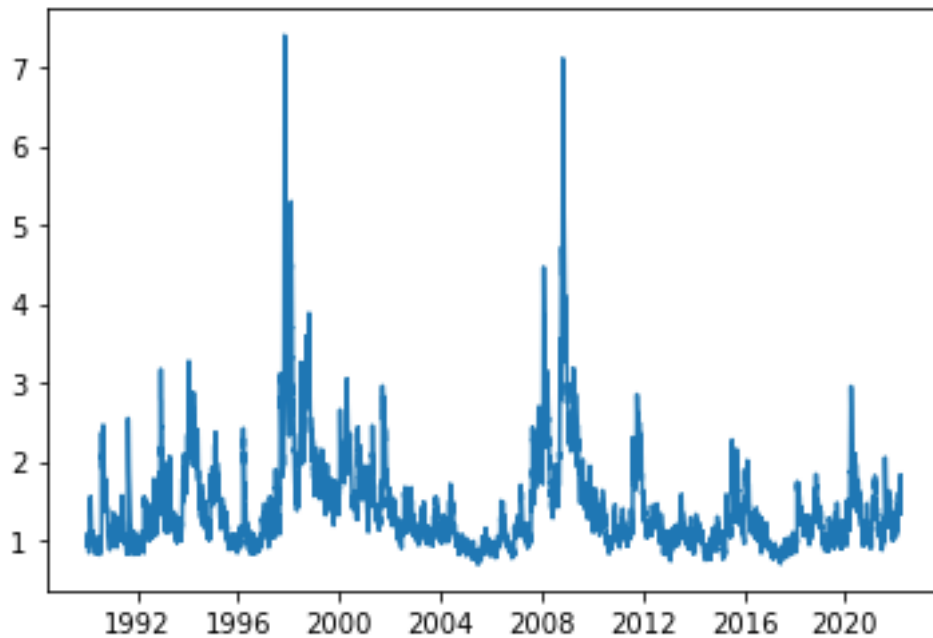


Figure 6.2.4 Plot of the volatility of the index

The remaining steps were to fit the model and generate a forecast using the LSTM model.

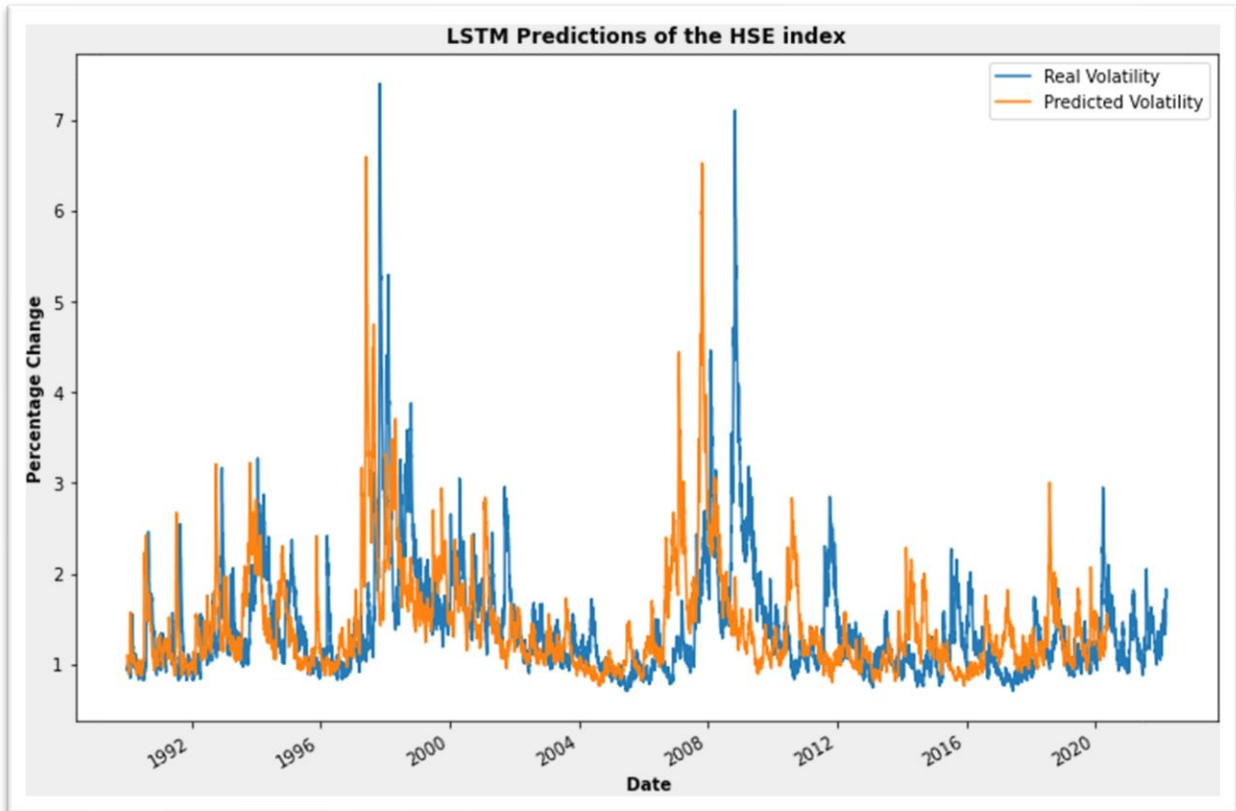


Figure 6.2.4 Plot of the prediction of the volatility of the index using LSTM
For more detail about the implementation, refer to Appendix C.

7 RESULTS

After calculating the Root Mean Squared Error, we obtained a value of 12% when it comes to the LSTM model. This means that only 12% of the predictions were wrong compared to the actual value. This may be due to the reduced number of epochs used to train the model which is a result of inferior processing power.

For the GARCH model, the results were accurate since the plot of the predicted volatility coincides with the percentage change of the returns of the stock index and its spikes. There is no need to measure the accuracy using metrics like the RMSE since the most important in predicting the volatility is to identify periods of high risk in a certain asset in the market and the value of the volatility itself is not important. It is considered an indicator instead of being a precise metric.

8 LIMITATIONS AND FUTURE WORK

The work that has been achieved has led me to conclude that the performance of the models could be enhanced using additional processing power, so this is a way that needs to be explored further.

Another alternative would be to enhance the efficiency of the computing power using parallel computing and conducting further research in this area.

CONCLUSION

Deep Learning is one of the most sophisticated branches of Artificial Intelligence that is applied in data science. This branch of research employed neural networks to simulate the way the brain operates in order to improve the learning of a model and provide more accurate results. The LSTM model is one of several neural network designs.

Financial engineering and quantitative finance emerged as a result of the expansion of both the financial and technical sectors in the financial industry. This area uses cutting-edge technology and sophisticated mathematical models to try to make sense of financial data.

Quantitative finance is now required in the financial business, whether it is to maximize the value of a portfolio, enhance wealth, or hedge against risk. In this industry, a true race has formed between numerous financial organizations, as old procedures are unable to keep up with the rate of change.

This study employed the GARCH model, one of the most advanced financial models for predicting volatility, and the LSTM model, a deep learning algorithm, to try to forecast the volatility of the Hong Kong stock index, the Hang Seng stock index, which includes prominent businesses listed in Hong Kong.

After making the predictions, we obtained valid results that were confirmed both analytically using the plot observations and the statistical data. This leads us to conclude that these two methods are indeed both accurate in predicting the volatility of a given asset, although the experiment showed the need for high computing power in order to achieve superior results compared to the more conventional GARCH model.

10 REFERENCES

- [1] S. Bragg, “Volatility definition,” AccountingTools, 14-Apr-2022. [Online]. Available: <https://www.accountingtools.com/articles/volatility>. [Accessed: 2022].
- [2] T. Green, “Stock market volatility explained,” The Motley Fool, 21-Dec-2020. [Online]. Available: <https://www.fool.com/investing/how-to-invest/stocks/stock-market-volatility/>. [Accessed: Apr-2022].
- [3] Fidelity International. [Online]. Available: <https://www.fidelity.com.sg/beginners/your-guide-to-stock-investing/understanding-stock-market-volatility-and-how-it-could-help-you>. [Accessed: Apr-2022].
- [4] D. Caplinger, “What is a stock market index? defined and listed,” The Motley Fool. [Online]. Available: <https://www.fool.com/investing/stock-market/indexes/>. [Accessed: Apr-2022].
- [5] Moneycontrol.com, “Hang Seng, Hang Seng Stock/share, Hang Seng index/HSI, hang seng live, hang seng market - asian market live, hong kong stock market,” English. [Online]. Available: <https://www.moneycontrol.com/live-index/hangseng>. [Accessed: Apr-2022].
- [6] A. byM. T. R. byD. Vaidya, A. byM. Thakur, A. by, M. Thakur, R. byD. Vaidya, R. by, and D. Vaidya, “Financial engineering,” WallStreetMojo, 12-Apr-2022. [Online]. Available: <https://www.wallstreetmojo.com/financial-engineering/>. [Accessed: Apr-2022].
- [7] Geek University, “Python overview: Python#,” Geek University. [Online]. Available: <https://geek-university.com/python-overview/>. [Accessed: Apr-2022].
- [8] K. Reyes, “What is deep learning and how does it works [updated],” Simplilearn.com, 21-Feb-2022. [Online]. Available: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning>. [Accessed: Apr-2022].
- [9] “Regression analysis,” Corporate Finance Institute, 29-Jul-2021. [Online]. Available: <https://corporatefinanceinstitute.com/resources/knowledge/finance/regression-analysis/>. [Accessed: Apr-2022].
- [10] “11.1 arch/GARCH models: Stat 510,” PennState: Statistics Online Courses. [Online]. Available: <https://online.stat.psu.edu/stat510/lesson/11/11.1>. [Accessed: Apr-2022].
- [11] “Generalized autoregressive conditional heteroskedasticity (GARCH) definition,” Investopedia, 11-Jun-2021. [Online]. Available: <https://www.investopedia.com/terms/g/garch.asp#:~:text=GARCH%20is%20a%20statistical>

%20modeling,an%20autoregressive%20moving%20average%20process. [Accessed: Apr-2022].

[12] M. Phi, “Illustrated guide to LSTM's and GRU's: A step by step explanation,” Medium, 28-Jun-2020. [Online]. Available: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>. [Accessed: Apr-2022].

[13] “Machine learning (in finance),” Corporate Finance Institute, 15-Jan-2022. [Online]. Available: <https://corporatefinanceinstitute.com/resources/knowledge/other/machine-learning-in-finance/>. [Accessed: Apr-2022].

[14] “What is numpy?” What is NumPy? - NumPy v1.22 Manual. [Online]. Available: <https://numpy.org/doc/stable/user/whatisnumpy.html>. [Accessed: Apr-2022].

[15] “What is python? executive summary,” Python.org. [Online]. Available: <https://www.python.org/doc/essays/blurb/>. [Accessed: Apr-2022].

11 APPENDIX A: CAPSTONE SPECIFICATION

MAIZI Sami

CSC

STOCK PRICE INDEX VOLATILITY FORECAST WITH LSTM AND GARCH

LAAYOUNI L

Spring 2022

The purpose of this capstone project is to forecast the volatility of a stock price index using a hybrid model that integrates both LSTM and multiple GARCH-type models.

The process will follow a process consisting of Analysis, Design, Implementation, and Testing steps. During the Analysis, a more elaborate list of project objectives and requirements will be made, and the scope and socio-economic implications of the project will be defined, in addition to an evaluation of how feasible the project is. During the Design phase, we will load the necessary time-series data to work on and choose the appropriate tools and resources to make the project. The implementation and testing will both go along as the results from testing will help improve or reimplement the project.

As for the project's timeline, the initial feasibility study will be conducted within a week from the official start of the project. The following month will be dedicated to the Analysis and the start of the Implementation where an Interim Report will assess the achieved progress at the time. The remaining two months will witness the continuation of implementation and the beginning of the Testing until reaching the final tests after achieving the goals set at the beginning. A final report will be made to give an account of the making process, and a final presentation will be given to present the project to the supervisors and coordinators for review. A total of three months will be dedicated to this project from the start of February until the end of April 2022.

To achieve a minimum degree of success, the project will need to provide accurate predictions of the stock index to be chosen using LSTM and GARCH models, in a format that will allow for decision making. The project must be achieved within the specified timeframe, and its respective deliverables must comply with the format and criteria of professionalism set by the school through the coordinator and supervisor of the project.

As the project will use technological tools, and used in a financial setting, in addition to influencing decision making, it is crucial to go over its societal and ethical implications. The predictions made using this tool could help reduce the riskiness related to investing. Decision-makers in the financial world are responsible for billions of dollars of their client's money, so they need to equip themselves with the most advanced tools to reduce risk, especially during these times of high uncertainty. Improving the profitability of investments in the stock market can be an incentive to deviate from all sorts of unethical investments, and needless to say that the making process as well as the investment products we will deal with complying with ethical rules.

12 APPENDIX B: FEASIBILITY STUDY

MAIZI Sami

CSC

STOCK PRICE INDEX VOLATILITY FORECAST WITH LSTM AND GARCH

LAAYOUNI L. and AZZOUZ M.

Spring 2022

This deliverable is a feasibility study for a capstone project that aims to forecast the volatility of a stock price index using a hybrid model that integrates both LSTM and multiple GARCH-type models.

The most basic tool that will be used for this project is my personal computer. It's a 7th gen i5 processor with 8 GB of Ram which is enough for the operations that will be conducted.

As a resource for the data, I will work with, I will rely both on YahooFiance.com and Investing.com, depending on the adequacy offered by each platform. The data will be in the .csv format, so I will have to use Excel, which is available on my computer under the university's license. I have used this tool extensively during MTH 3303, one of the uses being part of portfolio optimization in finance.

For the analysis, I will mainly use python – which I have first encountered in CSC 3323- and the variety of libraries and tools it offers like Numpy, matplotlib, pandas, Keras, and TensorFlow for Machine learning. As machine learning computations require extensive computing units, I will use Google Colab to run my program on the Cloud.

An additional resource that I will use is the book Hands-On Deep Learning for Finance by Troiano L. which contains many insights on the applications of Deep Learning in the financial sector.

I will reach out to my supervisors if I need any help or guidance since both Finance and Deep Learning require advanced mathematical knowledge.

Overall, after assessing the resources required to undergo this project, I can conclude that it is feasible and I can move on to further steps.

13 APPENDIX C: SNAPSHOTS OF THE CODE OF THE PROGRAM

```
import pandas_datareader.data as web
from datetime import datetime, timedelta
import pandas as pd
import matplotlib.pyplot as plt
from arch import arch_model
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import numpy as np

%matplotlib inline

import seaborn as sns
from matplotlib.colors import ListedColormap

import pandas_datareader.data as web
from datetime import datetime, timedelta
import pandas as pd
import matplotlib.pyplot as plt
from arch import arch_model
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import numpy as np
import cProfile

cp = cProfile.Profile()
cp.enable()

start = datetime(1990, 1, 1)
end = datetime(2022, 3, 10)

hse = web.DataReader('^HSI', 'yahoo', start=start, end=end)

returns = 100 * hse.Close.pct_change().dropna()

plt.figure(figsize=(10,4))
plt.plot(returns)
plt.ylabel('Pct Return', fontsize=16)
plt.title('HSE Returns', fontsize=20)

plot_pacf(returns**2)
plt.show()

model = arch_model(returns, p=2, q=2)

model_fit = model.fit()

model_fit.summary()
```

```

rolling_predictions = []
test_size = 365*20

for i in range(test_size):
    train = returns[:-(test_size-i)]
    model = arch_model(train, p=2, q=2)
    model_fit = model.fit(dis='off')
    pred = model_fit.forecast(horizon=1)
    rolling_predictions.append(np.sqrt(pred.variance.values[-1,:][0]))

rolling_predictions = pd.Series(rolling_predictions,
index=returns.index[-365*20:])

plt.figure(figsize=(10,4))
true, = plt.plot(returns[-365*20:])
preds, = plt.plot(rolling_predictions)
plt.title('Volatility Prediction - Rolling Forecast',
fontsize=20)
plt.legend(['True Returns', 'Predicted Volatility'],
fontsize=16)

train = returns
model = arch_model(train, p=2, q=2)
model_fit = model.fit(dis='off')

pred = model_fit.forecast(horizon=7)
future_dates = [returns.index[-1] + timedelta(days=i) for i in
range(1,8)]
pred = pd.Series(np.sqrt(pred.variance.values[-1,:]),
index=future_dates)

plt.figure(figsize=(10,4))
plt.plot(pred)
plt.title('Volatility Prediction - Next 7 Days', fontsize=20)

plt.plot(rolling_predictions)

cp.disable()
cp.print_stats()

x = model_fit._volatility
x = pd.DataFrame(x)
x = x.set_index(model_fit._index)
print(x.info())
ind = pd.to_datetime(model_fit._index)
x.set_index(ind)
plt.plot(x)
data = x

```

```

test_size = 12
test_index = len(data)- test_size

train = data.iloc[:test_index]
test = data.iloc[test_index:]

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(train)
scaled_train = scaler.transform(train)
scaled_test = scaler.transform(test)

from tensorflow.keras.preprocessing.sequence import
TimeseriesGenerator

length = 11
n_features=1
time_series_generator = TimeseriesGenerator(scaled_train,
scaled_train, length=length, batch_size=1)

X,y=time_series_generator[0]

X

scaled_train[:11]

print(y)
print(scaled_train[11])

initializer = tf.keras.initializers.he_uniform(seed=0)
model = Sequential()

model.add(LSTM(11, activation='relu', input_shape=(length,
n features),kernel initializer=initializer,

bias_initializer=initializers.Constant(0.01)))
opt = tf.keras.optimizers.Adam(learning_rate=0.001,
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-8)

model.compile(optimizer=opt, loss='mse')

time_series_val_generator =
TimeseriesGenerator(scaled_test,scaled_test, length=length,
batch_size=1)

from tensorflow.keras.callbacks import EarlyStopping

```

```

Early_Stopping =
EarlyStopping(monitor='val_loss',mode='min',verbose=1,patience=
10)

model.fit_generator(time_series_generator,epochs=20,
                    validation_data=time_series_val_generator ,
                    callbacks=[Early_Stopping])

loss = pd.DataFrame(model.history.history)
loss.plot()
plt.title('LSTM Training & Validation Loss',fontweight='bold')
plt.xlabel('Epochs',fontweight='bold')
plt.ylabel("Loss-'MSE'",fontweight='bold')
|
training_outputs = []
batch = scaled_train[:length].reshape((1, length, n_features))

for i in range(len(scaled_train[length:])):
    train_out = model.predict(batch)[0]
    training_outputs.append(train_out)
    batch =
np.append(batch[:,1:,:],[[scaled_train[length:][i]]],axis=1)

actual_train=scaled_train[length:]
actual_train=scaler.inverse_transform(actual_train)

train_predictions=scaler.inverse_transform(training_outputs)

plt.figure(figsize=(10,6))
plt.plot(actual_train)

plt.plot(train_predictions)
plt.title('LSTM Training Performance - Actual vs. Predicted
Training Values',fontweight='bold')
plt.legend(('Actual_Train_Values','Predicted_Train_Values'))
plt.xlabel('Training Time Steps 1-97',fontweight='bold')
plt.ylabel('Average MonthlyMinTemp',fontweight='bold');

train_err=abs((actual_train-
train_predictions)/actual_train)*100
train_err=pd.DataFrame(train_err,columns=['Training Error'])
plt.figure(figsize=(10,6))
sns.kdeplot(train_err['Training
Error'],shade=True,color='r',kernel='gau',)
plt.xlabel('Percentage of Training Relative
Error',fontweight='bold')
plt.title('Kernel Density Estimation ',fontweight='bold');

train_err.describe().transpose()

```

```

test_outputs = []
batch = scaled_train[-length:].reshape((1, length, n_features))

for i in range(len(test)):
    test_out = model.predict(batch)[0]
    test_outputs.append(test_out)
    batch = np.append(batch[:,1:,:], [[test_out]], axis=1)

lstm_predictions = scaler.inverse_transform(test_outputs)

test['LSTM Predictions'] = lstm_predictions

test

test.plot(figsize=(10,5));
plt.title('LSTM 12-Month Prediction - Actual vs. Predicted
Values',fontweight='bold')
plt.ylabel('Average MonthlyMinTemp',fontweight='bold')
plt.xlabel(' Monthly Time Steps',fontweight='bold');

train_index=pd.date_range(start='1990-1-
1',periods=7927,freq='B')

train_df=pd.DataFrame(data=train_predictions,index=train_index,
columns=['Predicted_Train'])

ax=data.plot(figsize=(12,8), label = 'Real Values')



train_df.plot(ax=ax, label = 'Prediction')
plt.title('LSTM Predictions of the HSE
index',fontweight='bold')
plt.ylabel('Percentage Change',fontweight='bold')
plt.xlabel(' Date',fontweight='bold');
plt.gca().legend(('Real Volatility','Predicted Volatility'))

from sklearn.metrics import mean_squared_error
print(np.sqrt(mean_squared_error(test[0],test['LSTM
Predictions'])))

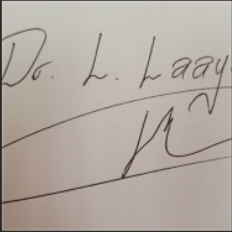
```

14 APPENDIX D: EMAIL APPROVALS

Re: Updated Final Report Approval

 **Lahcen Laayouni** <L.Laayouni@aui.ma>
4/28/2022 10:07 AM 

To: Sami Maizi <77780>; Mohamed Azzouz



Dear Sami,

Thank you for your email. I approve the report, please find enclosed my signature,

With best regards,
Laayouni

From: Sami Maizi <77780> <S.Maizi@aui.ma>
Sent: Thursday, April 28, 2022 12:07 AM
To: Mohamed Azzouz <Mo.Azzouz@aui.ma>; Lahcen Laayouni <L.Laayouni@aui.ma>
Subject: Updated Final Report Approval

Dear Supervisors,

Could you please approve my Updated Final Report by sending me your signatures ?

Sincerely,

Sami Maizi.

RE: Updated Final Report Approval



Mohamed Azzouz <Mo.Azzouz@au.ma>

4/28/2022 3:05 PM



To: Sami Maizi <77780>; Lahcen Laayouni

Dear Sami,

I approve the report.

Mohamed Azzouz

From: Sami Maizi <77780> <S.Maizi@au.ma>

Sent: 28 April 2022 00:08

To: Mohamed Azzouz <Mo.Azzouz@au.ma>; Lahcen Laayouni <L.Laayouni@au.ma>

Subject: Updated Final Report Approval

Dear Supervisors,

Could you please approve my Updated Final Report by sending me your signatures ?

Sincerely,

Sami Maizi.