PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
Mohamed Khider University of Biskra
Faculty of Exact Sciences and Sciences of Nature and Life

Department of Computing Sciences

Order N°: RTIC20/M2/2022

Dissertation

Presented to obtain the academic master degree in

Computer Science

Path: **Networks and Technologies of Information and Communication (RTIC)**

Deep learning for multivariate data. Application to "Digital Twin"

By: Assassi Abderrahim

Defended on 26/06/2022 in front of jury composed of:

Ayed Soheyb	MCA	President
Terrissa Sadek Labib	Professor	Reporter
Chami Djazia	Doctor	Examiner

Academic Year: 2021-2022

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة محمد خيضر بسكرة

تصريح شرفي

(خاص بالالتزام بقواعد النزاهة العلمية لإنجاز بحث)

أنا الممضي أسفله،

السيد (ة): ....عساسي عبد الرحيم ....الصفة: ....**طـــالب**.......................

الحامل لبطاقة التعريف الوطنية رقم: ....200497553....والصادرة بتاريخ: 28 | 09 | 2016

المسجل بكلية: العلوم الدقيقة و علوم الطبيعة و الحياة

قسم: الإعــــلام الآلـي

والمكلف بإنجاز مذكرة تخرج في الماستر عنوانها:

....

Deep learning for multivariate data. Application to "Digital Twin"

....

أصرح بشرفي أني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات المهنية والنزاهة الأكاديمية المطلوبة في إنجاز البحث المذكور أعلاه.

التاريخ: .... 22| 06 | 2022 ....

توقيع المعني:

[illegible]

Dedication

All gratitude to Allah the Almighty

Who granted me the power to accomplish this work.

With a delicate heart filled with enormous love & appreciation,

I dedicate this Dissertation first & foremost

To my beloved parent,

To my dear sister, brothers, nephews,

To my beloved wife,

Who have showed me nothing but Love & Compassion

To my best friends,

Lastly, to all my family and friends who have been the source of all the support

& happiness in this journey.

Acknowledgements

I express sincere appreciation towards all the people who made this work possible.

Profound gratitude goes to my supervisor Mr. Sadek Labib Terrissa for his continuous assistance and valuable guidance.

I thank members of the Jury for their efforts in evaluating this work.

Lastly, I deeply thank all the teachers for their constant support and advice throughout this journey.

Abstract

Digital twin has become recently an important concept in industry and academia. Make a digital representation of a system or a physical assets help to improve processes, study the expected behaviour of production and maximize benefits.

Many problems make the digital twin model not accurate to the real or physical asset, one of these problems is Data Drift. In this thesis we will make a system to resolve inaccurate of a model due to the data drift problem.

The system detects the drift in data using statistics calculation. we calculate mean and variance for each features in each data initial and new (online), then we calculate the percentage of change between the two calculation results. We update to the digital twin basing the calculation results.

Keywords: Digital twin, multivariate data, data drift, Statistics, deep learning model.

ملخص

أصبح التوأم الرقمي مؤخرًا مفهومًا مهمًا في الصناعة والأوساط الأكاديمية .يساعد التمثيل الرقمي لنظام أو الأصول المادية على تحسين العمليات ودراسة السلوك المتوقع للإنتاج و زيادة الفوائد.

العديد من المشاكل تجعل نموذج التوأم الرقمي غير دقيق بالنسبة للأصل الحقيقي أو المادي ، ومن بين هذه المشاكل انجراف البيانات . في هذه الأطروحة ، سنقوم بإنشاء نظام لحل مشكلة عدم دقة النموذج بسبب مشكلة انحراف البيانات.

يكتشف النظام الانحراف في البيانات باستخدام حساب الإحصائيات. نحسب المتوسط والتباين لكل ميزة في كل من البيانات الأولية و البيانات الجديدة ، ثم نحسب النسبة المئوية للتغيير بين نتيجتي الحساب. نقوم بتحديث نتائج الحساب إلى التوأم الرقمي.

الكلمات المفتاحية: توأم رقمي ، بيانات متعددة المتغيرات ، انجراف البيانات ، الإحصاء ، نموذج التعلم العميق.

Résumé

Le jumeau numérique est devenu récemment un concept important dans l'industrie et le milieu universitaire. Faire une représentation numérique d'un système ou d'un actif physique permet d'améliorer les processus, d'étudier le comportement attendu de la production et de maximiser les bénéfices.

De nombreux problèmes rendent le modèle de jumeau numérique non précis par rapport à l'actif physique, l'un de ces problèmes est le data drift. Dans cette thèse, nous allons créer un système pour résoudre les inexactitudes d'un modèle en raison du problème de data drift.

Le système détecte la dérive des données à l'aide de calculs statistiques. Nous calculons la moyenne et la variance pour chaque caractéristique dans chaque données initiale et nouvelle données, puis nous calculons le pourcentage de changement entre les deux résultats de calcul. Nous mettons à jour le jumeau numérique en nous basant sur les résultats des calculs.

Mots-clés : Jumeau numérique, données multivariées, data drift, statistiques, modèle d'apprentissage en profondeur.

Table of Contents

CHAPTER 6 : IMPLIMENTATION AND RESULTS

List of Figures

List of Source Code

List of Tables

GENERAL INTRODUCTION

General Introduction

The idea of digital twin technology was first voiced in 1991, with the publication of Mirror Worlds, by David Gelernter. Digital twin has become recently an important concept in industry and academia. Digital twin defined as a virtual model or representation of a system that can be used to understand performance better, improve processes or create revenue opportunities from services [4].

Digital Twins can become a tool, unconstrained by domains, beyond the boundaries of high performing economic regions, and contribute to economic growth, through open source platforms for digitization. The impact from Digital Twins, and benefits, may be only limited, by our imagination.

As we know the reliability of data make the decision more accurate, for that the data collected in time make our model as similar as reality, but the data can be altered or changed due to many causes. The data drift is one of problems we will concentrate in this thesis.

Statistically, dataset drift between a source distribution S and a target distribution T is defined by the change in the joint distribution of features and target.

Deep learning is a machine-learning approach based on algorithms that attempt to model high-level abstractions in data by using a deep neural network. Deep learning typically uses artificial neural networks, which are mathematical models inspired by the structure and function of the human brain. Deep learning is used in different fields, Deep learning technology lies behind everyday products and services such as digital assistants as well as emerging technologies such as self-driving cars.

In this thesis we aim to concentrate on data drift problem, for that we design a system to detect the data drift using a statistics analyses of data then we create the digital twin using a deep learning model.

This thesis is divided into 6 chapters, in the first chapter we present the concept of digital twin and the domain of application, its importance, its types and we finish with an example of conceptual architecture.

In the second chapter, we detailed the problem of data drift and we see the causes of it, types of data drift and we finish with methods to detect and handle it.

In the third chapter we talk about statistical data analyses, we begin with definitions, types of data, types of statistics analysing, then we detailed the data analyses process. In the fourth chapter we present the deep learning concept.

The fifth chapter is the design of the detection system, wish is implemented in the sixth and final chapter.

CHAPTER 1: DIGITAL TWIN

Introduction

The digital twin concept was formally introduced by professor Michael Grieves at a Society of Manufacturing Engineers conference. However, it was NASA who first embraced the digital twin concept. in a 2010, John Vickers of NASA introduced a new term “Digital Twin”. The idea was used to create digital simulations of space capsules and craft for testing.
Digital twin has become recently an important concept in industry and academia. Make a digital representation of a system or a physical assets help to improve processes, study the expected behaviour of production and maximize benefits. In this Chapter we will talk about digital twin concept, importance, types …

1. History

The concept of digital twins was first put forward by David Gelernter’s 1991 book ‘Mirror Worlds,’ with Michael Grieves of the Florida Institute of Technology going on to apply the concept to manufacturing [3].

The use of the “twin/twins” concept in the manufacturing can be traced back to NASA’s Apollo program. In the project, NASA needs to make two identical spacecraft. The aircraft left on earth is called a twin and is used to reflect the status/condition of the space vehicle in action.

In 2003, professor Michael Grieves proposed the concept of virtual digital representations equivalent to physical products in a product lifecycle management (PLM) course at the University of Michigan and gives it a definition: which is a digital copy of one or a set of specific devices that can abstractly represent a real device and can be used as a basis for testing under real or simulated conditions [3].

In 2011, professor Michael Grieves, in his book “Virtually perfect: driving innovative and lean products through product lifecycle management”, cited the conceptual model of the noun digital twin model (digital twin), which is described by his co-author John Vickers, and it is still in use today. Its conceptual model includes three main parts: (1) real space entity products; (2) virtual space virtual products; and (3) data and information interface between real space and virtual space.

In 2012, in the face of future aircraft light quality, high load and the demand of the longer service time under more extreme environment, NASA and the US Air Force Research Laboratory in cooperation put forward the common digital twin example of future aircraft. For aircraft, flight systems or launch vehicles, they define digital twin as an integrated multi-physical, multi-scale, probabilistic simulation model for aircraft or system, which uses the best available physical models, updated sensor data, and historical data to reflect the state of the flying entity corresponding to the model [3].

In 2014, Professor Michael Grieves elaborated digital twin in detail in his white paper “Digital Twin: Manufacturing Excellence through Virtual Factory Replication”. In the same year, the U.S. defence department, PTC, Siemens and Assaults accepted the term “digital twin” and began to use it in marketing campaigns. It needs to be pointed out that they all use “digital twin” instead of “digital twins”.

For two consecutive years (2016 and 2017), Gartner, the world’s most authoritative IT research and consulting firm, listed digital twin as one of the top 10 strategic technology trends of the year.

It can be seen that the digital twins have developed rapidly in both theoretical and application levels in recent years. At the same time, the application range has gradually shifted from the product design stage to the product manufacturing stage and operation and service stage, which has attracted wide attention of scholars and enterprises [3].

2. Definitions

2.1. Digital twin

There are a lot of definitions for digital twin; A Digital Twin is defined as a virtual representation of a connected physical asset. Or as A set of virtual information constructs that mimics the structure, context and behaviour of an individual/unique physical asset, or a group of physical assets, is dynamically updated with data from its physical twin throughout its life cycle and informs decisions that realize value [1].

It defined also as a virtual model or representation of a system such as product design, process or a factory that can be used to understand performance better, improve processes or create revenue opportunities from services [4].

A digital twin can be defined, fundamentally, as an evolving digital profile of the historical and current behaviour of a physical object or process that helps optimize business performance [8].

A digital twin is a virtual representation of an object or system that spans its lifecycle, is updated from real-time data, and uses simulation, machine learning and reasoning to help decision-making [9].

Based on the previous definitions, there are three crucial characteristics of Digital Twins, which should be included in the definition:

1) The Digital Twin is a virtual dynamic representation of a physical artefact or system,
2) Data is automatically and bi-directionally exchanged between the Digital Twin and the physical system,
3) The Twin entails data of all phases of the entire product lifecycle and is connected to all of them.

Consequently, the following definition was derived:
"A Digital Twin is a virtual dynamic representation of a physical system, which is connected to it over the entire lifecycle for bidirectional data exchange" [5].

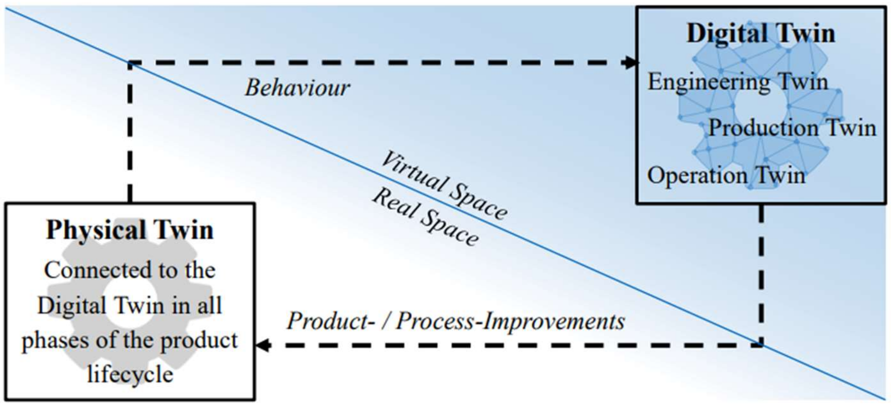


Fig. 1.1: Concept of digital twin [5]

The physical twin automatically transfers, data of its behaviour, status, and information on the environment from the real space to the virtual space over the entire product lifecycle, when needed. The virtual twin instead identifies product or process oriented improvements, control demands based on the current situation, or predictions of the near future and sends them back to the real space so the physical product adapts accordingly. This data exchange happens automatically (Fig. 1.1). Here, it is important to differentiate between data and information. The twins only exchange data and Information is only generated in the real. The virtual space based on the transmitted data so an automatic data exchange is required. When the Digital Twin needs to influence the physical object, data should be transferred on demand, only required data should be entailed, so that the Digital Twin can represent the Physical Twin sufficiently well [5].

2.2. Level of integration

According to the level of data integration, the Digital Twins can classify into three subcategories, Digital Model, Digital Shadow and Digital Twin [10].

2.2.1. Digital Model

A Digital Model is a digital representation of an existing or planned physical object that does not use any form of automated data exchange between the physical object and the digital object. The digital representation might include a more or less comprehensive description of the physical object. These models might include, but are not limited to simulation models of planned factories, mathematical models of new products, or any other models of a physical object, which do not use any form of automatic data integration. Digital data of existing physical systems might still be in use for the development of such models, but all data exchange is done in a manual way. A change in state of the physical object has no direct effect on the digital object and vice versa.

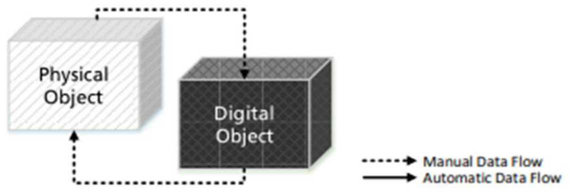


Fig. 1.2: Data Flow in a Digital Model [10]

2.2.2. Digital Shadow

Based on the definition of a Digital Model, if there further exists an automated one-way data flow between the state of an existing physical object and a digital object, one might refer to such a combination as Digital Shadow. A change in state of the physical object leads to a change of state in the digital object, but not vice versa.

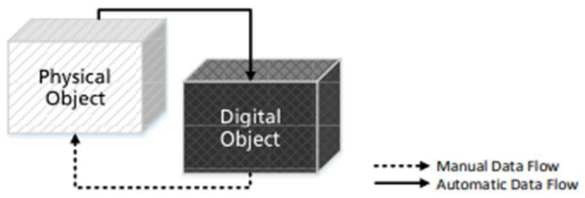


Fig. 1.3: Data Flow in a Digital Shadow [10]

2.2.3. Digital Twin

If further, the data flows between an existing physical object and a digital object are fully integrated in both directions, one might refer to it as Digital Twin. In such a combination, the digital object might also act as controlling instance of the physical object. There might also be other objects, physical or digital, which induce changes of state in the digital object. A change in state of the physical object directly leads to a change in state of the digital object and vice versa.

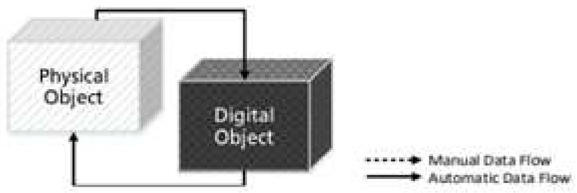


Fig. 1.4: Data Flow in a Digital Twin [10]

3. Digital Twin Applications

Digital twins are used in a wide variety of industries for a range of applications and purposes. The primary areas of interest are:

3.1. Manufacturing: The biggest reason for this is that manufacturers are always looking for a way in which products can be tracked and monitored in an attempt to save time and money, a key driver and motivation for any manufacturer.

The Digital Twin has the potential to give real-time status on machines performance as well as production line feedback. It gives the manufacturer the ability to predict issues sooner. Digital Twin use increases connectivity and feedback between devices, in turn, improving reliability and performance prediction analysis. The Digital Twin is creating an environment to test products as well as a system that acts on real-time data.

Another application of Digital Twin is in the automotive industry, most notably demonstrated by Tesla. The ability to have a Digital Twin of an engine or car part can be valuable in terms of using the twin for simulation and data analytics [2].

3.2. Smart cities: Digital twin can also be used to help cities become more economically, environmentally and socially sustainable. Virtual models can guide planning decisions and offer solutions to the many complex challenges faced by modern cities.

The ability of services and infrastructures within a smart city to have sensors and to be monitored with IoT devices is of great value for all kinds of future-proofing. It can be used to help in the planning and development of current smart cities and help with the ongoing developments of other smart cities. As well as the benefits of planning, there are also benefits within the energy saving world. This data gives an excellent insight into how our utilities are being distributed and used [2].

Advancement for the smart city is the potential to utilize Digital Twin technology. It can facilitate growth by being able to create a living tested within a virtual twin that can achieve two things; one, to test scenarios, and, two, to allow for Digital Twins to learn from the environment by analysing changes in the data collected.

3.3. Healthcare: The increased connectivity is only growing the potential application of Digital Twin use within the healthcare sector. One future application is a Digital Twin of a human, giving a real-time analysis of the body.

A more realistic current application is a Digital Twin used for simulating the effects of certain drugs. Another application sees the use of a Digital Twin for planning and performing surgical procedures [11].

Having the ability to simulate and act in real-time is even more essential in healthcare domain and it can be the difference between life or death. The Digital Twin could also assist with predictive maintenance and ongoing repair of medical equipment. The Digital Twin within the medical environment has the potential along with AI to make life saving decisions based on real-time and historical data [2].

3.4. Automotive: The automobile industry has been improved by digital twin technology. Digital twins in the automobile industry are implemented by using existing data in order to facilitate processes and reduce marginal costs. Currently, automobile designers expand the existing physical materiality by incorporating software-based digital abilities [13].

A specific example of digital twin technology in the automotive industry is where automotive engineers use digital twin technology in combination with the firm's analytical tool in order to analyse how a specific car is driven, in purpose to suggest new features in the car that can reduce car accidents on the road, this was not possible in the short past time [12].

4. Importance of Digital Twin

Why we might want a digital twin? The benefits of digital twin differ depending when and where it is used. For example, using digital twin to monitor existing products, such as a wind turbine or oil pipeline, can reduce maintenance charge and save millions in costs. Digital twins can also be used for prototyping ahead of manufacture, reducing product defects and shortening time to market.

Other instances of digital twin use can include process improvements, whether that is monitoring of staffing levels against output or aligning a supply chain with manufacturing or maintenance requirements.

Common benefits include increased reliability and availability through monitoring and simulation to improve performance. They can also reduce risk of accidents and unplanned downtime through failure, lower maintenance costs through predicting failure before it occurs, and ensure production goals are not impacted by scheduling maintenance, repair and the ordering of replacement parts. Digital twin can also offer continued improvements by analysing customization models and ensure product quality through performance testing in real time.

By imitate physical assets, frameworks and operations to produce continuous data, a digital twin allows industry to prevent downtime, react to changing circumstances, test design improvements, Understand the present stat and prevent the future, detect problems before they occur, develop new opportunities and Plan “what if” scenarios using simulation.

5. Types of Digital twin

Michael Grieves and John Vickers divided Digital Twins into two types Digital Twin Prototype (DTP) and Digital Twin Instance (DTI) [7]. Another type can be added witch is Digital Twin Aggregate (DTA).

- Digital Twin Prototype (DTP): This is undertaken before a physical product is created, this type of Digital Twin describes the prototypical physical artifact. It contains the informational sets necessary to describe and produce a physical version that duplicates or twins the virtual version. These informational sets include, but are not limited to, Requirements, Fully annotated 3D model, Bill of Materials (with material specifications), Bill of Processes, Bill of Services, and Bill of Disposal.

- Digital Twin Instance (DTI): This is done once a product is manufactured in order to run tests on different usage scenarios, this type of Digital Twin describes a specific corresponding physical product that an individual Digital Twin remains linked to throughout the life of that physical product. Depending on the use cases required for it, this type of Digital Twin may contain, but again is not limited to, the following information sets: A fully annotated 3D model with Geometric Dimensioning and Tolerancing (GD&T) that describes the geometry of the physical instance and its components, a Bill of Materials that lists current components and all past components, a Bill of Process that lists the operations that were performed in creating this physical instance, along with the results of any measurements and tests on the instance, a Service Record that describes past services performed and components replaced, and Operational States captured from actual sensor data, current, past actual, and future predicted.

- Digital Twin Aggregate (DTA): This gathers DTI information to determine the capabilities of a product, run prognostics and test operating parameters.

DT's are operated on in a Digital Twin Environment (DTE) this is an integrated, multi-domain physics application space for operating on Digital Twins for a variety of purposes. These purposes would include:

- Predictive: The Digital Twin would be used for predicting future behaviour and performance of the physical product. At the Prototype stage, the prediction would be of the behaviour of the designed product with components that vary between its high and low tolerances in order to ascertain that the as-designed product met the proposed requirements. In the Instance stage, the prediction would be a specific instance of a specific physical product that incorporated actual components and component history. The predictive performance would be based from current point in the product's lifecycle at its current state and move forward. Multiple instances of the product could be aggregated to provide a range of possible future states.
- Interrogative: This would apply to DTI's. Digital Twin Instances could be interrogated for the current and past histories. Irrespective of where their physical counterpart resided in the world, individual instances could be interrogated for their current system state: fuel amount, throttle settings, geographical location, structure stress, or any other characteristic that was instrumented. Multiple instances of products would provide data that would be correlated for predicting future states. For example, correlating component sensor readings with subsequent failures of that component would result in an alert of possible component failure being generated when that sensor pattern was reported. The aggregate of actual failures could provide Bayesian probabilities for predictive uses.

6. Digital Twin life cycle

• As designed, representing the intended structure and performance of the real world entity
• As built, representing the output of the manufacturing and assembly process
• As used and maintained, representing the status of the physical object in operation

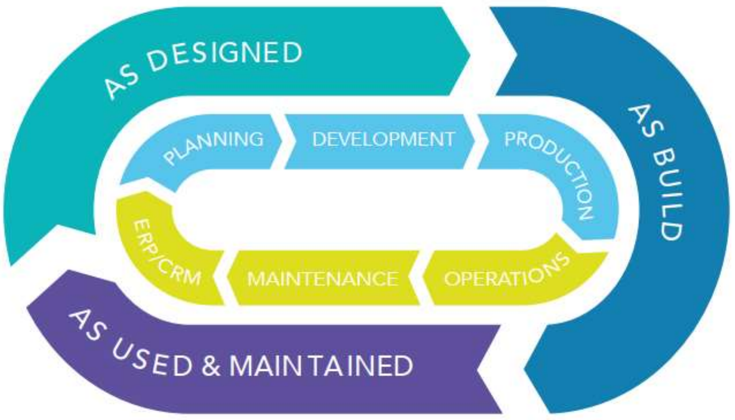


Fig. 1.5: Digital Twin life cycle [4]

7. Digital twin conceptual architecture

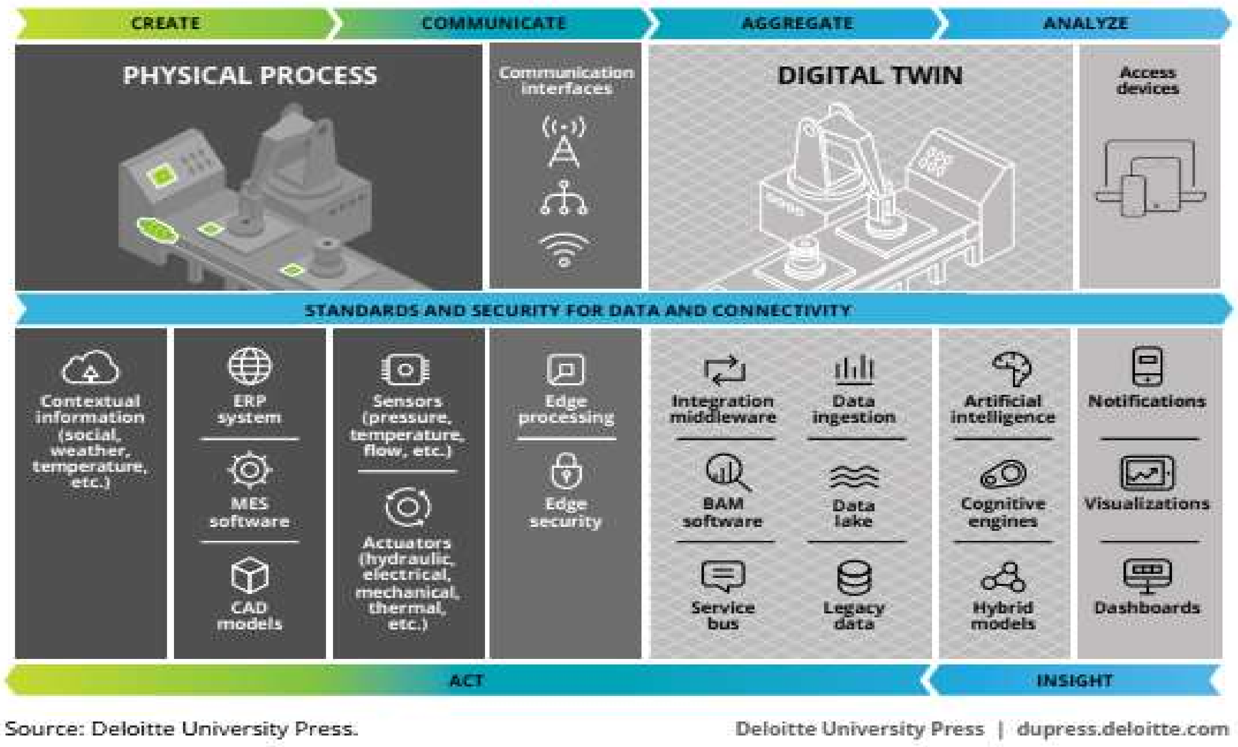


Fig. 1.6: Digital win conceptual architecture [8]

7.1. Create: encompasses outfitting the physical process with myriad sensors that measure critical inputs from the physical process and its surroundings.

7.2. Communicate: helps the seamless, real-time, bidirectional integration/connectivity between the physical process and the digital platform.

7.3. Aggregate: can support data ingestion into a data repository, processed and prepared for analytics.

7.4. Analyse: In this step, data is analysed and visualized. Data scientists and analysts can utilize advanced analytics platforms and technologies to develop iterative models that generate insights and recommendations and guide decision making.

7.5. Insight: In this step, insights from the analytics are presented through dashboards with visualizations, highlighting unacceptable differences in the performance of the digital twin model and the physical world analogue in one or more dimensions, indicating areas that potentially need investigation and change.

7.6. Act: The act step is where actionable insights from the previous steps can be fed back to the physical asset and digital process to achieve the impact of the digital twin.

Conclusion

The growth in Digital Twin use has seen a shift in recent years, facilitated by an increase in the number of published papers and industry leaders investing heavily in developing Digital Twin technology.

In the near future, many of us will be living with a digital twin. This is an internet-connected virtual representation of ourselves that can do anything we can do. It's predictably hard to believe in something like this just yet, but soon it could be an everyday reality. And they could make our lives a lot better.

In this chapter we talked about the important concepts for digital twin.

CHAPTER 2: DATA DRIFT

Introduction

The explosion of data sources and rapid innovation in advanced analytics, data science, AI, and machine learning has fundamentally changed the scale and pace of data integration. The importance of making decisions quickly in real time force the organization to be subordinate on fresh, reliable data than ever before.

The main concern of data scientists is the model pertinence over time. Is the model still capturing the pattern of new incoming data, and is it still performing as well as during its design phase? [14].

The major problem about the data accuracy is how detect the drift? , when we must change the model? And what we should do to change the model? to ensure its pertinence. In this chapter, We'll explain what data drift is, how it can affect your analysis.

1. Definitions

Data drift is unexpected and undocumented changes to data structure, semantics, and infrastructure that is a result of modern data architectures. Data drift breaks processes and corrupts data, but can also reveal new opportunities for data use [15].

Data drift also called covariate shift is a situation when the distribution of model inputs changes. Data drift is a shift in the distribution of the input features between training and serving data [16]. Data drift is one of the top reasons model accuracy degrades over time. For machine learning models, data drift is the change in model input data that leads to model performance degradation. Monitoring data drift helps detect these model performance issues [19].

It can arise in multiple ways. When the data is collected by some sensor, the device can break down or receive a software update that impacts how the measurements are taken. Or When the data is on human beings, it can change along with fashion or user demographics.

Statistically, dataset drift between a source distribution S and a target distribution T is defined by the change in the joint distribution of features and target.

$$P(xs\ , ys) \neq P(xt\ , yt)$$

2. Causes of data drift

Typical causes of data drifts include:

- Upstream process changes, such as a sensor being replaced that changes the units of measurement from inches to centimeters.
- Data quality issues, such as a broken sensor always reading 0.
- Natural drift in the data, such as mean temperature changing with the seasons.
- Change in relation between various features, or covariate shift.

3. Drift types

When talking about drift, several different closely related terms may come up: covariate shift, concept drift, data drift, model drift, model decay, dataset shift, distribution shift. These terms refer to different assumptions about what's changing [20].

3.1. Covariate Shift (Shift in the independent variables):

Covariate shift is the change of distributions in one or more of the independent variables (input features). This means that due to some environmental change even though the relationship between feature X and target Y remains unchanged, the distribution of feature X has changed. The graph below may help understand this better.

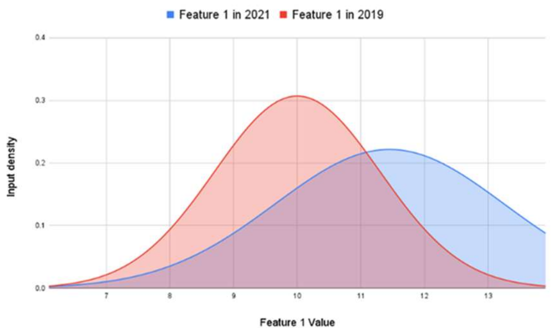


Fig. 2.1: Example of Covariate Shift

The example above: Due to the pandemic many businesses closed or their revenues decreased, they had to reduce staff, etc, but they decided to keep paying their loans because they were afraid that the bank may take seize their assets. Performance degradation will be more apparent when this sort of shift happens in one or more of the top contributing variables of a model [20].

Covariate shift is the situation where:

Ptrain(Y|X) = Ptest(Y|X) but Ptrain(X) ≠ Ptest(X)

Where *Ptest* could be your test set of data after the model has been deployed.

3.2. Prior Probability Shift (Shift in the target variable):

With prior probability shift, the distribution of the input variables remains the same but the distribution of the target variable changes. For example, that could look something like this:

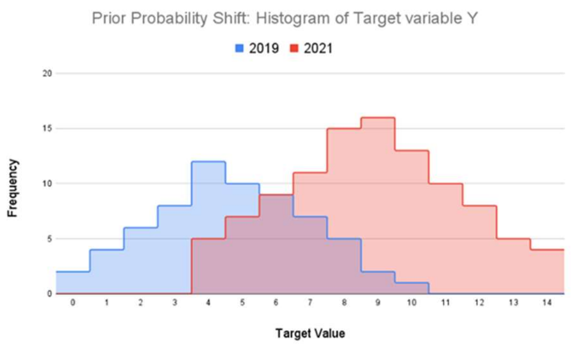


Fig. 2.2: Example of Prior Probability Shift

Companies that were not really affected by the lockdown and have not suffered any revenue losses but deliberately chose not to repay their loan instalments to take advantage of government subsidies and maybe save that money in case the situation does worsen for them in the future (same X distribution but different Y) [20].

Prior Probability Shift is the situation where:

Ptrain(X|Y)=Ptest(X|Y) but Ptrain(Y) ≠Ptest(Y)

Where *Ptest* could be your test set of data after the model has been deployed.

3.3. Concept Shift

With concept drift the relationships between the input and output variables change. This means that the distributions of input variables (such as user demographics, frequency of words, etc.) might even remain the same and we must instead focus on the changes in the relationship between X and Y.

In more formal definition terms, concept shift is the situation where:

$$Ptrain(Y|X) \neq Ptest(Y|X) \text{ and } Ptrain(X) = Ptest(X) \text{ in } X\ Y$$
$$Ptrain(X|Y) \neq Ptest(X|Y) \text{ and } Ptrain(Y) = Ptest(Y) \text{ in } Y\ X$$

Where *Ptest* could be your test set of data after the model has been deployed.

Concept drift is more likely to appear in domains that are dependent on time, such as time series forecasting and data with seasonality. Learning a model over a given month won't generalize to another month.

There are a few different ways in which concept drift might show up:

3.3.1. Gradual Concept Drift

Gradual or incremental drift is the concept drift that we can observe over time and therefore expect. With different changes in the world, our model gradually becomes outdated resulting in a gradual decline in its performance [20].

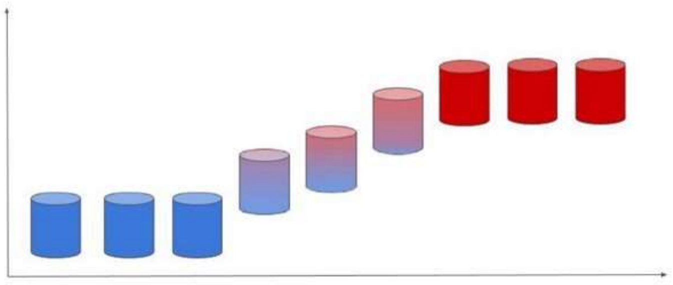


Fig. 2.3: Gradual Concept Drift

Some examples of gradual concept drift are:

Launch of alternative products: products that weren't available during the training period (for example if the product was the only one of its kind in the market) can cause unforeseen effects on the model since it has not seen similar trends before
Economic changes: changes in interest rates and maybe its effect on more loan borrowers to default on their loans can cause changes.
The effect of situations like these can add up over time to cause a more dramatic drift effect.

3.3.2. Sudden Concept Drift

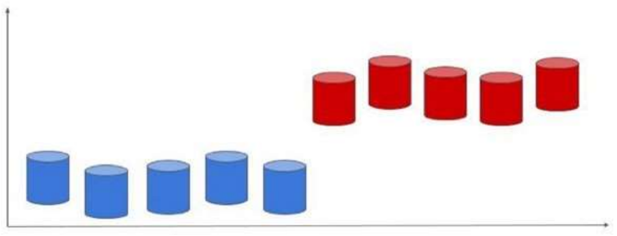


Fig. 2.4: Sudden Concept Drift

As the name suggests, these concept shifts happen by surprise and suddenly. Some of the most apparent examples came when COVID-19 first struck on a global scale. Demand forecasting models were heavily affected, supply chains couldn't keep up [20].

But such changes can also happen during the regular function of a company when there is no pandemic. In general, any major change in the environment that throws the model into unfamiliar territory will cause performance degradation.

3.3.3. Recurring Concept Drift

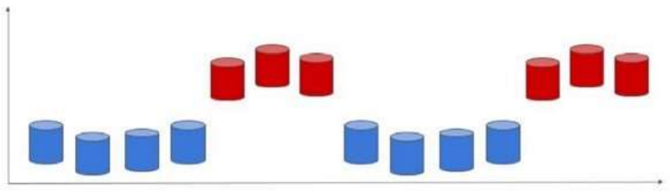


Fig. 2.5: Recurring Concept Drift

Recurrent concept drift is pretty much "seasonality". But seasonality is common in machine learning with time series data and is something we are aware of. So if we expect this sort of drift, for example, a different pattern on weekends or certain holidays of the year, we just need to make sure that we train the model with data representing this seasonality. This sort of data drift usually becomes a problem in production only if a new pattern develops that the model is unfamiliar with [20].

4. Detect Data Drift

Drift is a key issue because machine learning often relies on a key assumption: the past == the future. In the real world, this is very rarely the case. As a result, it's critical to understand how changes in the data will affect the model's behavior both before a model is deployed and on an ongoing basis during deployment [14].

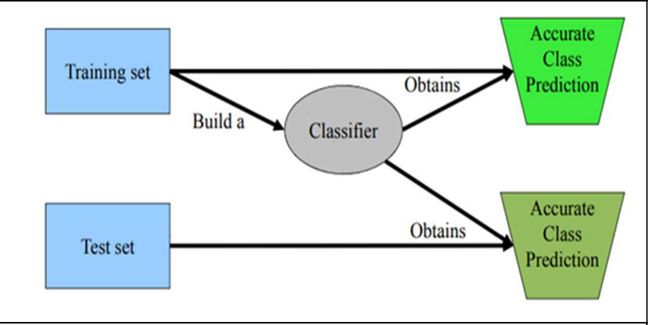
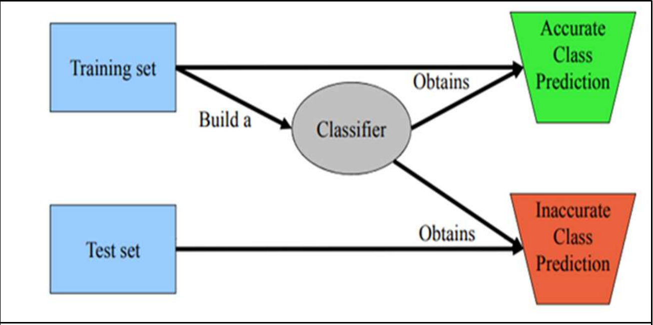
 Training set Build a Classifier Obtains Accurate Class Prediction Test set Obtains Accurate Class Prediction	 Training set Build a Classifier Obtains Accurate Class Prediction Test set Obtains Inaccurate Class Prediction
Expected outcome	*Outcome with hits of data drift*

Fig. 2.6: Data drift in outcome [14]

When we see cases like the one in the Fig. 2.6 the model gets inaccurate predictions with bad accuracy, where the outputs had taken a significant turn. In parallel situations important questions must be asked: where in our data the data drift is occurring?, What in the data is causing this shift?, is the quality of my model stay accurate and How does this affect it?, Do I need to change my model?

Drift detection constitutes an important stage of the ML Model Lifecycle for flawless ML performance in production environments. A common approach to detecting data drift includes comparing training and production data sets distributions using a nonparametric test.

- **Methods for detection of Data Drift**

All the methods for detecting data drift are lagging indicators of drift. Only after they have processed enough data after any kind of drift that has occurred, that the actual drift is detected. [18] Some of the Following methods are a number of techniques that have been explored in python using the '*scikit-multiflow*' library [17].

4.1. Population Stability Index (PSI):

It compares the distribution of the target variable in the test dataset to a training data set that was used to develop the model [18].

$$PSI = \sum \left((Actual\% - Expected\%) \times ln\left(\frac{Actual\%}{Expected\%}\right) \right)$$

Steps for calculation:

1) Divide the expected (test) dataset and the actual (training dataset) into groups and define the boundary values of the groups based on the minimum and maximum values of that column in train data.

2) Calculate the % of observations in each group for both expected and actual datasets.

3) Calculate the PSI as given in the formula

3). a) When PSI<=1

This means there is no change or shift in the distributions of both datasets.

3). b) 0.1< PSI<0.2

This indicates a slight change or shift has occurred.

3). c) PSI>0.2

This indicates a large shift in the distribution has occurred between both datasets.

4.2. Kolmogorov-Smirnov Windowing method (KSWIN):

KSWIN (Kolmogorov-Smirnov Windowing) is a concept change detection method based on the Kolmogorov-Smirnov (KS) statistical test. KS-test is a statistical test with no assumption of underlying data distribution. KSWIN can monitor data or performance distributions. Note that the detector accepts one dimensional input as array. [17]

4.3. Page-Hinkley method (PageHinkley):

This change detection method works by computing the observed values and their mean up to the current moment. Page-Hinkley won't output warning zone warnings, only change detections. The method works by means of the Page-Hinkley test. In general lines it will detect a concept drift if the observed mean at some instant is greater than a threshold value lambda [17].

4.4. Model-Based Approach

A Machine Learning-based model approach can also be used to detect data drift between two populations.
We need to label our data which has been used to build the current model in production as 0 and the real-time data gets labeled as 1. We now have to build a model and evaluate the results.
If the model gives high accuracy, it means that it can easily discriminate between the two sets of data. Thus, we could conclude that a covariate shift has occurred and the model will need to be recalibrated. On the other hand, if the model accuracy is around 0.5, it means that it is as good as a random guess. This means that a significant data shift has not occurred and we can continue to use the model.
The disadvantage of this model is that every time new input data is made available, the training and testing process needs to be repeated which can become computationally expensive [18].

4.5. Adaptive Windowing(ADWIN):

ADWIN (ADaptive WINdowing) is an adaptive sliding window algorithm for detecting change, and keeping updated statistics about a data stream. ADWIN allows algorithms not adapted for drifting data, to be resistant to this phenomenon. The general idea is to keep statistics from a window of variable size while detecting concept drift. This algorithm is implemented in python for a specialized tool which can perform drift detection [17].

4.6. Drift Detection Method (DDM):

DDM (Drift Detection Method) is a concept change detection method based on the PAC learning model premise, that the learner's error rate will decrease as the number of analyzed samples increase, as long as the data distribution is stationary.

If the algorithm detects an increase in the error rate, that surpasses a calculated threshold, either change is detected or the algorithm will warn the user that change may occur in the near future, which is called the warning zone [17].

4.7. Early Drift Detection Method (EDDM):

EDDM (Early Drift Detection Method) aims to improve the detection rate of gradual concept drift in DDM, while keeping a good performance against abrupt concept drift.

This method works by keeping track of the average distance between two errors instead of only the error rate. For this, it is necessary to keep track of the running average distance and the running standard deviation, as well as the maximum distance and the maximum standard deviation [17].

5. Relate Works

There are many related works talking about data drift, in this section we present some articles:

Jian-Wei Liao and Bi-Ru Dai in their article [21] "An Ensemble Learning Approach for Concept Drift", they categorized Concept drifts briefly into sudden and gradual concept drifts. However, in the real world, a data stream probably has more than one type of concept drift, and the type is usually difficult to be identified. In light of these reasons, they proposed a new weighting method for the classification of streaming data with concept drift which can adapt more quickly to current concept.
The proposed method, called *Accuracy and Growth rate updated Ensemble* (AGE), was designed based on the geometric mean of weights and growth rates of models. With the proposed method, they have more flexibility on reacting to various types of concept drifts. The experiments showed that their method outperforms AUE2 on most of datasets.

Another method made by Geoffrey I. Webb an al in their article [57] "Analysing concept drift and shift from sample data", they propose a new data mining task, *concept drift mapping* — the description and analysis of instances of concept drift or shift. they argue that concept drift mapping is an essential prerequisite for tackling concept drift and shift. They propose tools for this purpose, arguing for the importance of quantitative descriptions of drift and shift in marginal distributions. They present quantitative concept drift mapping techniques, along with methods for visualizing their results. They illustrate their effectiveness for real-world applications across energy-pricing, vegetation monitoring and airline scheduling.

6. Handling data drift in production

In production, there are multiple ways to respond to data drift Some of the methods which are generally followed in the industry are [18]:

a. Blindly update model:

This is a naïve approach. There is no proactive drift detection. Models are periodically retrained and updated with recent data. Without drift detection in place, it is difficult to estimate the time interval for re-training and model re-deployment.

b. Training with weighted data:

When a new model is trained instead of discarding old training data, use weight inversely proportional to the age of data.

c. Incremental learning:

As new data arrives, the models are continuously retrained and updated. As a result, the model is always adapting to the changes in the data distribution. This approach will work with machine learning models which allow incremental learning one instance of data at a time.

Conclusion

In practice, identifying the exact type of data drift is important, but what matters more is identifying the impact on model performance and catching the drift on time so that actions such as retraining the model can be taken early on.

Model Monitoring and Drift Detection is an important part of the ML Model Lifecycle which needs to be optimized for successful and efficient deployments of models into production. Identifying any kind of drifts in the data in real-time and a proper strategy to handle such drifts is very crucial for the models to give better results with time.

In this chapter we see data drift definitions, types of drift, how to detect it and ways to handle it.

CHAPTER 3: STATISTICAL ANALYSIS OF DATA

Introduction

Statistical analysis of data refers to the extraction of some useful knowledge from vague or complex data. Statistical analysis of data includes importing, cleaning, transformation, etc. of data in preparation for analysis.

So in this chapter we will present the important concepts for statistical analysis of data, we will start with definitions, types of data, types of Statistics Analysing, then the data analysis process and we will conclude with the benefits of statistical analysis of data.

1. Definition

Instead of starting with the definition of statistics, Karl Pearson define it as: “Statistics is the grammar of science”. [23]

Statistics (or statistical analysis) is the process of collecting and analysing data to identify patterns and trends. It's a method of using numbers to try to remove any bias when reviewing information. It can also be thought of as a scientific tool that can inform decision making. [22]

Recently everyone is talking about Data. After hearing the word “Data” the basic questions that arise in our minds are, What is Data? , How Data is collected? , How Data can be analysed? , How Data is interpreted?

To answer all these questions, the term “Statistics” is used. Statistics is the basic and important tool to deal with the data. Now coming to the definition of statistics, it involves the collection, descriptive, analysis and concludes the data [23].

2. Types of Data

Univariate, bivariate and multivariate are the various types of data that are based on the number of variables. Variables mean the number of objects that are under consideration as a sample in an experiment. Usually there are three types of data sets.

2.1. Univariate data

Univariate data is used for the simplest form of analysis. It is the type of data in which analysis are made only based on one variable. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it. The example of a univariate data can be height [54].

Heights (in cm)	164	167.3	170	174.2	178	180	186

Fig 3.1: Univariate data example [54]

2.2. Bivariate data

Bivariate data is used for little complex analysis than as compared with univariate data. Bivariate data is the data in which analysis are based on two variables per observation simultaneously. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship among the two variables. The example of bivariate data can be temperature and ice cream sales in summer season [54].

TEMPERATURE(IN CELSIUS)	ICE CREAM SALES
20	2000
25	2500
35	5000
43	7800

Fig 3.2: Bivariate data example [54]

2.3. Multivariate data

Multivariate data is the data in which analysis are based on more than two variables per observation. Usually multivariate data is used for explanatory purposes.

It is similar to bivariate but contains more than one dependent variable. The ways to perform analysis on this data depends on the goals to be achieved. Some of the techniques are regression analysis, path analysis, factor analysis and multivariate analysis of variance (MANOVA) [54].

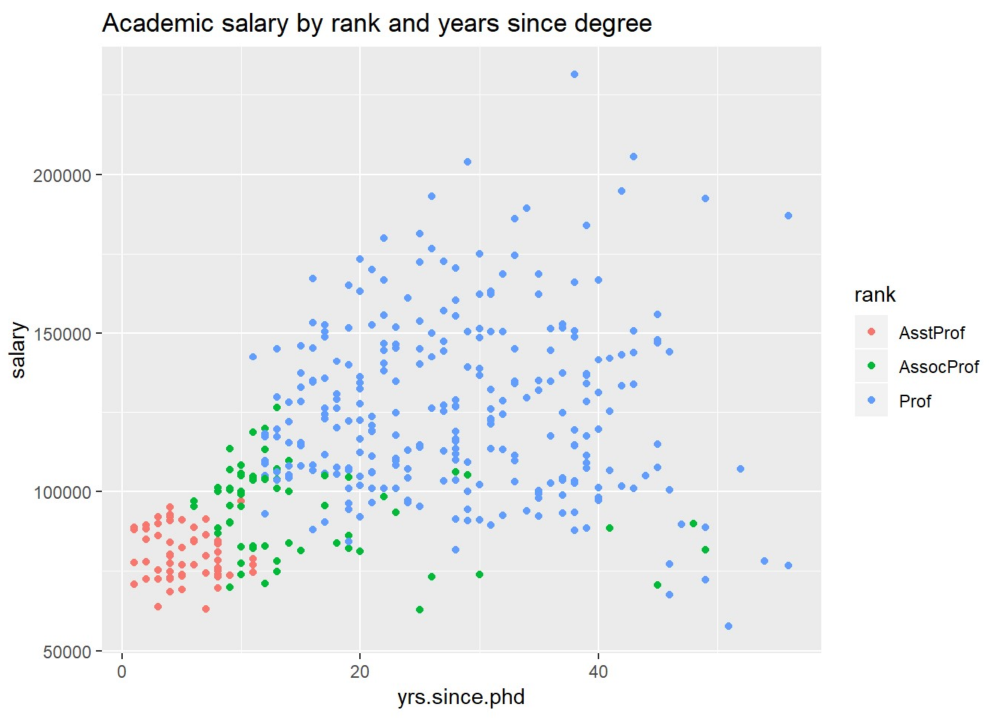


Fig 3.3: Multivariate data example [55]

3. Types of Statistics Analysing

There are two types of Statistics, Descriptive and Inferential Statistics.

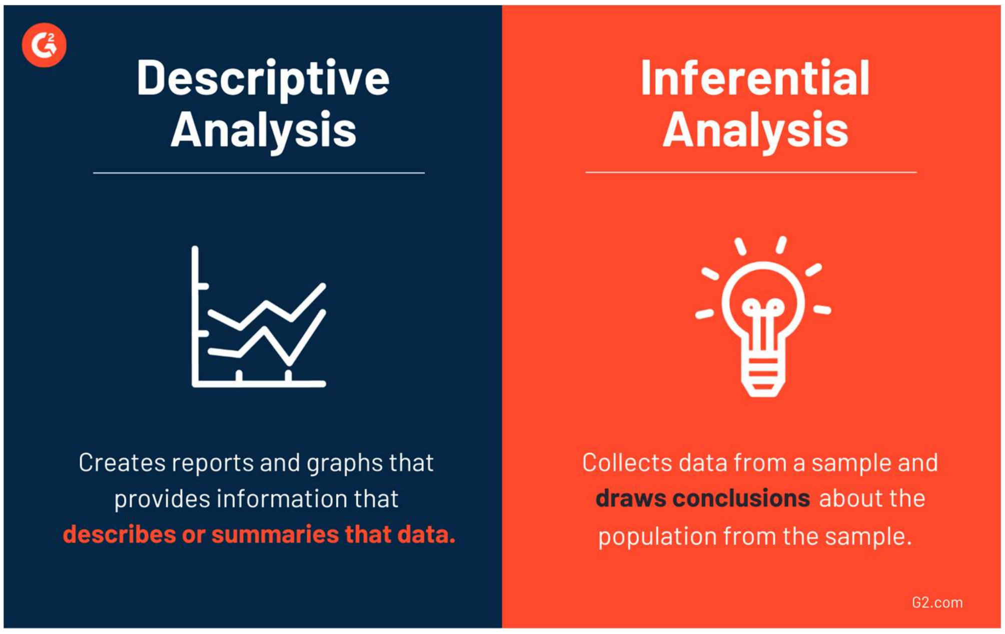

Descriptive Analysis
Creates reports and graphs that provides information that describes or summaries that data.
Inferential Analysis
Collects data from a sample and draws conclusions about the population from the sample.
G2.com

Fig 3.4: Difference between Descriptive and Inferential Statistics

3.1. Descriptive Statistics

In Descriptive Statistics, from the given observation the data is summarized. Descriptive statistics enables us to present the data in a more meaningful way, which allows simpler interpretation of the data. It doesn't make any predictions about the data. Several methods are used to analyse descriptive statistics of data such as mode, median and mean, as well as range, variance and standard deviation [24].

There are four different categories in Descriptive Statistics. They are:

- Measure of frequency
- Measure of dispersion
- Measure of central tendency
- Measure of position.

Based on the number of times a particular data has occurred defines the measure of frequency. The Measure of dispersion can be defined based on the Range, Variance, Standard Deviation, etc., The mean, median, mode, skewness of the respective data comes under the measure of central tendency. Finally, based on the percentile and quartile the position is measured.

Descriptive statistics are limited in so much that they only allow you to make summations about the objects that you have actually measured. You cannot use the data you have collected to generalize to other objects (i.e., using data from a sample to infer the properties/parameters of a population). For example, if you tested a drug to beat cancer and it worked in your patients, you cannot claim that it would work in other cancer patients only relying on descriptive statistics (but inferential statistics would give you this opportunity) [24].

3.2. Inferential Statistics

looking into Inferential Statistics, once the data is collected, tabulated, and analysed the summary or the inference is derived by using inferential statistics. The inferences are drawn based upon sampling variation and observational error.

Based on the information and conclusion derived from the sample the inferential statistics help us to predict and estimate results for the population.

This type of statistical analysis is intended to extract inferences or hypotheses from a sample of large data. Prediction about the population is carried out from random samples of data.

The prediction of the dependent variable based on the independent variable is carried out in inferential statistics.

Some technical terms are used to make a prediction about sample data are listed below: [25]

- **Z-Score:** is a way to compute the probability of data occurring within the normal distribution. It shows the relationship of different values in data with the mean of data. To compute the Z score, we subtract the mean from each data value and divide the whole by standard deviation. Z score is computed for a column in the dataset. It tells whether a data value is typical for a specific dataset. Z score helps us to decide whether to keep or reject the null hypothesis. The null hypothesis refers that there is no spatial pattern among the data values associated with the features. Z score can be imported from "scipy" library of python.
- **Z-test:** is to analyse whether the means of two different samples of data are similar or different while knowing their variances and standard deviations. It is a hypothetical test that follows a normal distribution. It is used for large-size data samples. It tells if the two datasets are similar or not. In this case, the null hypothesis considers that both datasets are significantly similar. A significance level (say 5%) is to be set sot that the null hypothesis is only accepted if the p-value of data is more than the significant level. A good z-test signifies that both the dataset are similar and are not significantly different from each other. The z-test method can be implemented using the library called "statsmodels" in python.
- **T-test:** is also used to determine whether the two datasets are similar or different. It is the same as z-test but the difference is that this method is applicable to a smaller sample size. The T-test can be implemented using libraries like numpy, pandas, and scipy.
- **F-test:** utilizes F-distribution. It is used to determine if the two samples of data are equal based on comparing their variances. The null hypothesis is rejected if the ratio of the variances of two samples of data is equal to one. There is some significance level also to tolerate some amount of difference between the two samples which is not considered significant. It is implemented using "scipy" library of python.

There are two main limitations to the use of inferential statistics. The first, and most important limitation, which is present in all inferential statistics, is that we are providing data about a population that we have not fully measured, and therefore, cannot ever be completely sure that the values/statistics we calculate are correct. Remember, inferential statistics are based on the concept of using the values measured in a sample to estimate/infer the values that would be measured in a population; there will always be a degree of uncertainty in doing this. The second limitation is connected with the first limitation. Some, but not all, inferential tests require the user (i.e., us) to make educated guesses (based on theory) to run the inferential tests. Again, there will be some uncertainty in this process, which will have repercussions on the certainty of the results of some inferential statistics [24].

4. The data analysis process

Seth Godin said "**Data is not useful until it becomes information**"

Before getting into Analysing the data, there are few things to remember. Define your question, Collect the right data, Understand the data, Cleaning the data, Analyse the data and finally interpret the results for the questions.

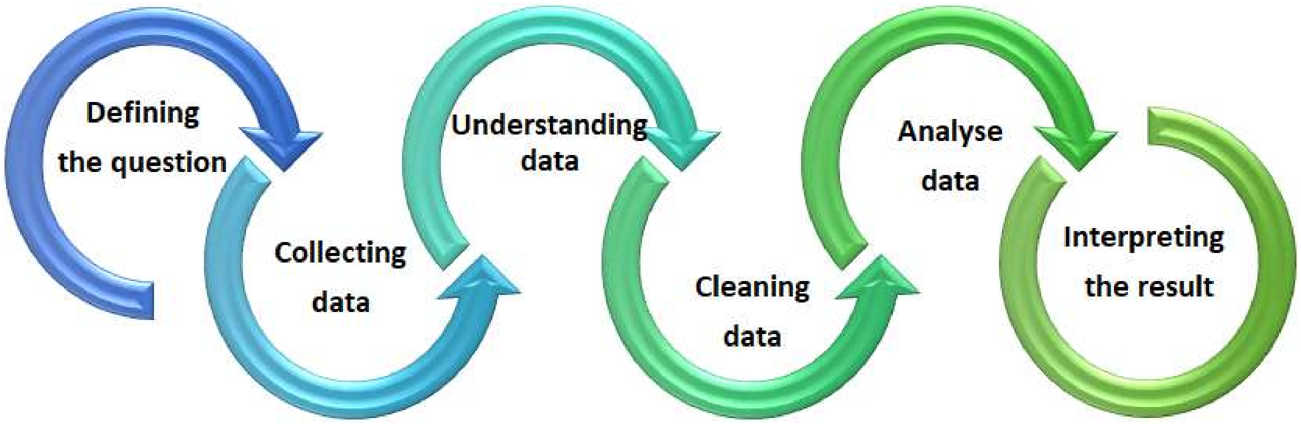


Fig 3.5: Data analysis process

4.1. Defining the question

For an organization, the betterment steps are taken from the past data analysis. For better steps, there will be a few objectives to be answered perfectly to give a good interpretation. The question should give the potential solution for the problem. For that framing, a relevant question is more important. Based on the questions only, the data will be collected. So defining the question is plays a major role [23].

4.2. Collecting data

The data collection has two classifications. One is primary data and another is secondary data. In primary data, the data will be collected through questionnaires, by sending emails or approaching each person. For example, census. Whereas, in secondary data, it's the data that is already available in the secondary source like Database [23].

Now before collecting the new data, identify the existing data that is available from the database. Apart from that collect the relevant data to satisfy the objective. Then organize the existing data with the new data to proceed with the analysis.

4.3. Understanding data

Once the data is collected there may be many variables that are related directly or indirectly to the objective. For that, we first need to study about all the variables whether it is nominal or ordinal. Preparing the data for analysis is done after understanding the data. While understanding we get to know about the data types, rows, and columns, missing in the data, finding the independent and dependent variables, etc.,

There may be few variables that may not be related to the question that the organization has and those variables can be used for future analysis. For finding those kinds of variables, understanding the data is more important [23].

4.4. Cleaning data

Data cleaning is the process of modifying the data, removing the duplicate variables, creating other variables if needed. Deleting the unwanted columns that are not related to the question. If the data cleaning is not proper it may lead to a lower accuracy of the model and may tend to misleading conclusions.

Once the data cleaning is done, the right data to answer the question is ready. Data manipulation is done in many ways like plotting the data, creating pivot tables for the variables, correlation, regression, detecting outliers process may take place. During the stage of manipulation, it may be needed to proceed with an existing dataset or remove some dataset or there may be a need to add few more data to answer the question. After all these stages, the required data will be ready for analysis [23].

4.5. Analyse data

When starting to talk about analysis the main thing is that model selection. Selecting the model plays an important role to analyse the data and answering the objective. Defining the dependent and independent variables is the important stage when analysing the data.

Currently, machine learning techniques are used for data analysis such that predictions and interpretations can be done easily. But still, some objectives can be answered directly while doing data visualization and basic statistical analysis. One of many tools and libraries used for analysing the data is Python [23].

4.6. Interpreting the result

After analysing the data, it's time to interpret the result. While interpreting the result check whether the analysis answered all the questions that were framed, does the data collected helped in the analysis, and from the interpretation is there a positive result for the betterment of the objective [23].

5. Benefits of Statistical Analysis of Data

Statistical analysis of data is important because it saves time and optimizes the problem. Statistical analysis is important to good decisions on data. Statistical analysis of data helps us to access effective data only with good efficiency. It helps us to decide an optimal path for data accessing and processing.

statistical analysis brings in numerous benefits to make the best usage of the vast data available, such as assisting in market research, product development, mapping out the company's growth rate, improve the efficiency of the company, etc.

Conclusion

Statistical analysis of data is the acquisition of knowledge about data in order to simplify the complex data which can be further used for processing. The goal of data analysis is to optimize the complex data structure. It helps us to take optimal decisions on data.

Statistical data analysis generally involves some form of statistical tools, which a non-expert cannot perform without having any statistical knowledge. There are various software packages to perform statistical data analysis. Python is one of many tools and libraries used for analysing the data.

CHAPTER 4: DEEP LEARNING

Introduction

Deep learning is an improved technique of machine learning. Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services such as digital assistants as well as emerging technologies such as self-driving cars.

Deep learning is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain, albeit far from matching its ability, allowing it to learn from large amounts of data. The layers are interlinked and each layer receives the output of the previous layer as input. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Neural Networks and Deep Learning

There are many application areas for Deep Learning, which covers such as Image Processing, Natural Language Processing, biomedical, Customer Relationship Management automation, Vehicle autonomous systems and others [35].

1. Artificial Neural Networks

Inspired by the sophisticated functionality of human brains where hundreds of billions of interconnected neurons process information in parallel, researchers have successfully tried demonstrating certain levels of intelligence on silicon. Examples include language translation and pattern recognition software [30].

ANNs are at the very core of Deep Learning. They are versatile, powerful, and scalable, making them ideal to tackle large and highly complex Machine Learning tasks such as classifying billions of images (e.g., Google Images), powering speech recognition services (e.g., Apple's Siri), recommending the best videos to watch to hundreds of millions of users every day (e.g., YouTube) [33].

1.1. Biological neurons

First, let’s take a quick look at a biological neuron. It is an unusual-looking cell mostly found in animal brains. It is composed of a cell body containing the nucleus and most of the cell’s complex components, many branching extensions called dendrites, plus one very long extension called the axon. The axon’s length may be just a few times longer than the cell body, or up to tens of thousands of times longer. Near its extremity the axon splits off into many branches called telodendria, and at the tip of these branches are minuscule structures called synaptic terminals (or simply synapses), which are connected to the dendrites or cell bodies of other neurons [33].

Biological neurons produce short electrical impulses called action potentials (APs, or just signals) which travel along the axons and make the synapses release chemical signals called neurotransmitters. When a neuron receives a sufficient amount of these neurotransmitters within a few milliseconds, it fires its own electrical impulses (actually, it depends on the neurotransmitters, as some of them inhibit the neuron from firing) [33].

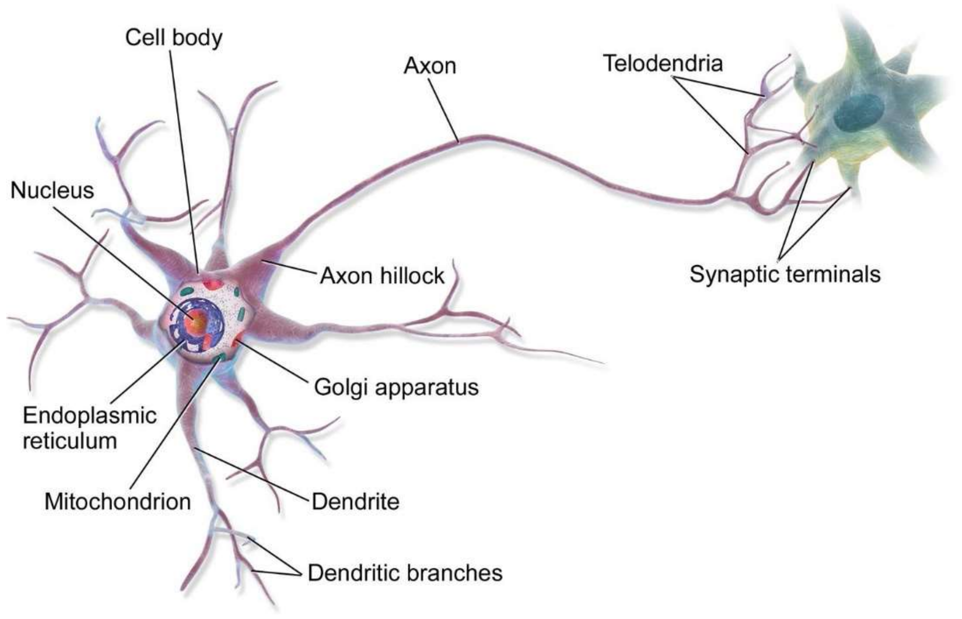


Fig 4.1: Biological neurons [33]

1.2. Artificial neurons

Artificial neuron is a basic building block of every artificial neural network. Its design and functionalities are derived from observation of a biological neuron that is basic building block of biological neural networks (systems) which includes the brain, spinal cord and peripheral ganglia.

Biological neuron and artificial neuron are Similar in design and functionalities. where the left side of a Fig 3.2 represents a biological neuron with its soma, dendrites and axon and where the right side of a figure represents an artificial neuron with its inputs, weights, transfer function, bias and outputs [31]

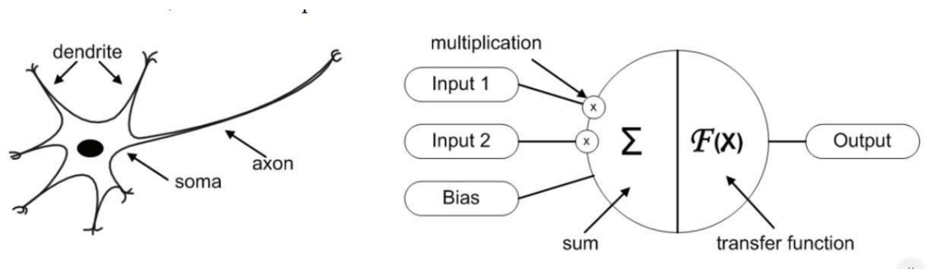


Fig 4.2: Biological and artificial neuron design [31]

In case of biological neuron, the information comes into the neuron via dendrite, soma processes the information and passes it on via axon. In case of artificial neuron, the information comes into the body of an artificial neuron via inputs that are weighted (each input can be individually multiplied with a weight) [28].

The body of an artificial neuron then sums the weighted inputs, bias and processes the sum with a transfer function. At the end an artificial neuron passes the processed information via output(s). Benefit of artificial neuron model simplicity can be seen in its mathematical description below [31]:

$$y(k) = F\left(\sum_{i=0}^{m} w_i(k) \cdot x_i(k) + b\right) \quad (3.1)$$

Where:

- $x_i(k)$ is input value in discrete time k where i goes from 0 to m.
- $w_i(k)$ is weight value in discrete time k where i goes from 0 to m.
- b is bias.
- F is a transfer function.
- $y(k)$ is output value in discrete time k.

- As seen from a model of an artificial neuron and its equation (3.1) the major unknown variable of our model is its transfer function. Transfer function defines the properties of artificial neuron and can be any mathematical function. We choose it on the basis of problem that artificial neuron (artificial neural network) needs to solve and in most cases we choose it from the following set of functions: binary step function, linear function and non-linear function (Sigmoid, tanh, ReLU, Leaky ReLU …).

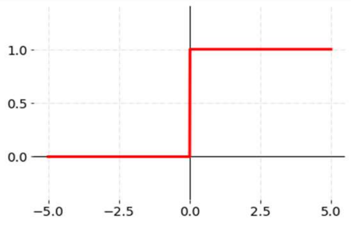
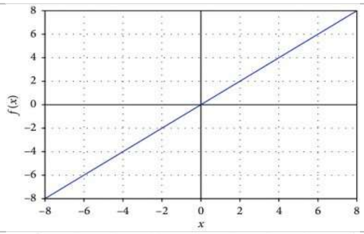
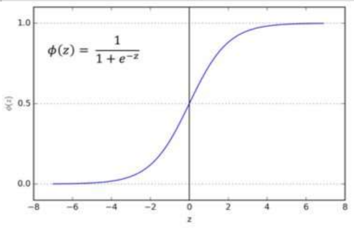
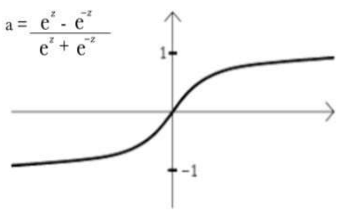
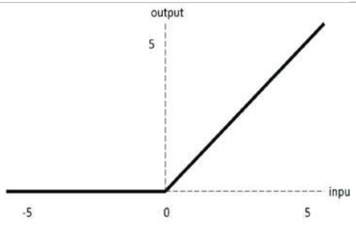
Fonction Activation	Equation	Graphe
Binary Step Function	$F(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$	
Linear Activation Function	$F(x) = a\,x$	
Sigmoid Function	$S(x) = \dfrac{1}{(1 + e^{-x})}$	
Tanh Function	$tanh(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	
ReLU Function	$RELU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$	

Table 4.1: Transfer function

1.3. Layers

In this classic artificial neural networks there are many types of layers used in the network, each type of layer is responsible for some computations [31].

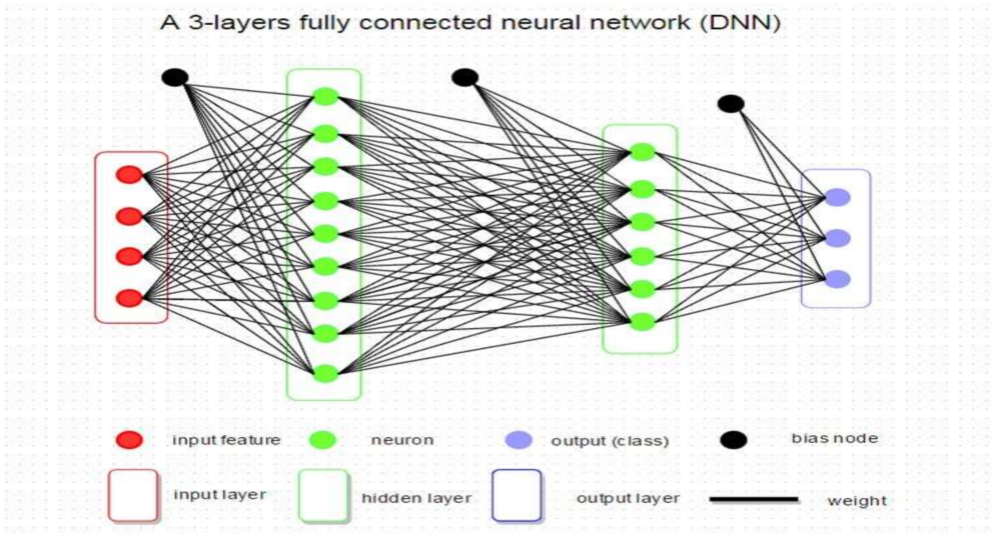


Fig 4.3: Artificial neural networks architecture [32]

- **Input layer:** Input layer is the first layer in the neural network, composed of input neurons and brings initial data to the hidden layers for further processing.
- **Hidden layer:** The layer or group of layers between the input and output layer. Deep learning is an optimization problem looks for the optimal solution of a very complex problem, many of these computations are made in the hidden layer(s). The choice of hidden layers depends on the complexity of the data, when using a less complex data it's recommended to use few hidden layers, using many hidden layers in a simple problem can lead to overfitting, and using simple architecture in complex problem leads to under-fitting. Each hidden layer of the network is typically vector-valued. The dimensionality of these hidden layers determines the width of the model.
- **Output layer:** Output layer is the last layer in neural network which produces the outputs of the program, in classification tasks, the size of output layer is equal to number of classes [32].

2. Deep neural network architectures

2.1. Convolutional neural networks

Convolutional neural networks or CNNs in short, are the popular choice of neural networks for different Computer Vision tasks such as image recognition. The name convolution is derived from a mathematical operation involving the convolution of different functions [33].

Convolutional networks have been tremendously successful in practical applications. The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers [32].

There are 4 primary layers or stages in designing a CNN:

- **Convolution:** The input signal is received at this layer.
- **Subsampling or Pooling:** Inputs received from the convolution layer are smoothened to reduce the sensitivity of the filters to noise or any other variation.
- **Activation:** This layer controls how the signal flows from one layer to the other, similar to the neurons in our brain.
- **Fully connected:** In this stage, all the layers of the network are connected with every neuron from a preceding layer to the neurons from the subsequent layer [33].

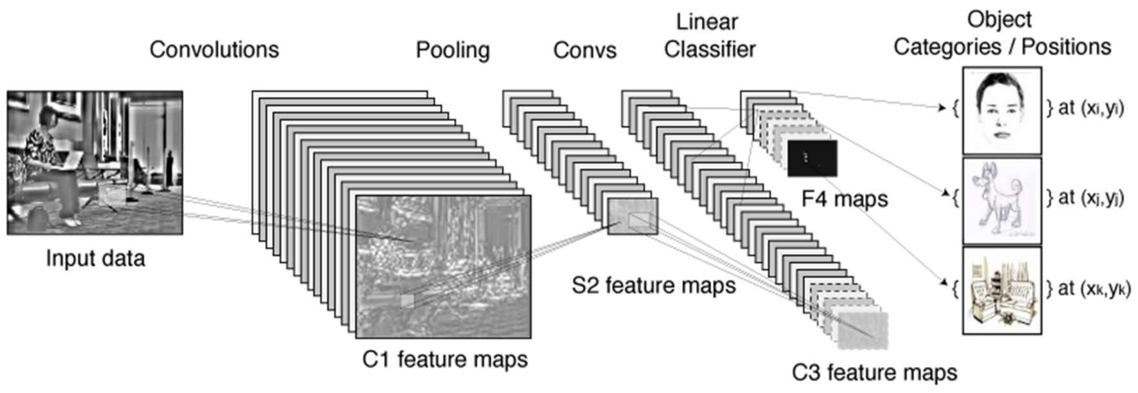
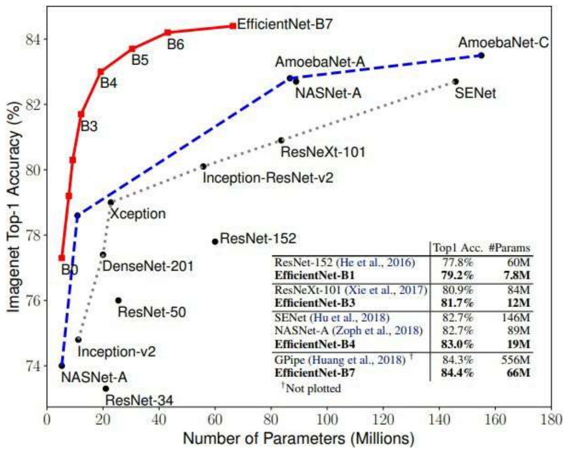


Fig 4.4: CNN architecture for visual recognition [34]

Advantages of CNN	Disadvantages of CNN
- Very good for visual recognition. - Once a segment within a particular sector of an image is learned, the CNN can recognize that segment present anywhere else in the image [35].	- CNN is highly dependent on the size and quality of the training data. - Highly susceptible to noise [36].

Table 4.2: Advantages and disadvantages of CNN [35] [36]

There are various architectures of CNNs available which have been key in building algorithms which power and shall power AI as a whole in the foreseeable future. Some of them: LeNet, AlexNet, ResNet, EffecientNet, and InceptionNet.



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Fig 4.5: ImageNet Top-1 accuracy CNN models [41]

2.2. Recurrent neural networks

Recurrent Neural Networks (RNNs) have been very popular in areas where the sequence in which the information is presented is crucial. As a result, they find a lot of applications in real-world domains such as natural language processing, speech synthesis and machine translation.

RNNs are called recurrent mainly because a uniform task is performed for every single element of a sequence, with the output dependent on the previous computations as well. Think of these networks as having a memory, where every calculated information is captured, stored and utilized to calculate the final outcome [37].

Over the years, quite a few varieties of RNNs have been researched and developed:

- **Bidirectional RNN:** The output in this type of RNN depends not only on the past but also the future outcomes.
- **Deep RNN:** In this type of RNN, there are multiple layers present per step, allowing for a greater rate of learning and more accuracy.

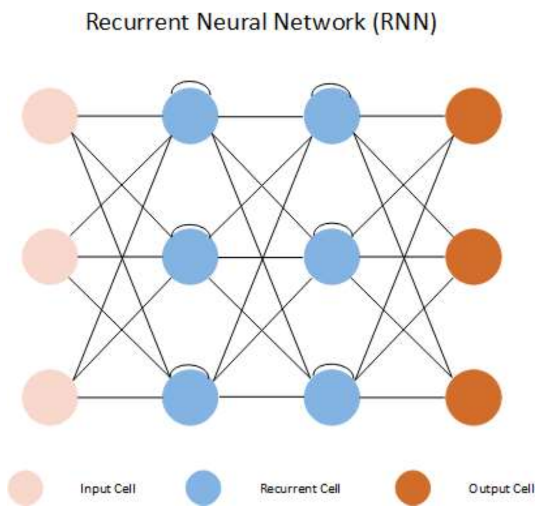


Fig 4.6: RNN architectures

Advantages of RNN	Disadvantages of RNN
- Unlike a traditional neural network, an RNN shares the same parameters across all steps. This greatly reduces the number of parameters that we need to learn. - RNNs can be used along with CNNs to generate accurate descriptions for unlabelled images.	- RNNs find it difficult to track long term dependencies. This is especially true in case of long sentences and paragraphs having too many words in between the noun and the verb. - RNNs cannot be stacked into very deep models. This is due to the activation function used in RNN models, making the gradient decay over multiple layers.

Table 4.3: Advantages and disadvantages of RNN [38]

2.3. Long Short-Term Memory

Long Short-Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people. They work tremendously well on a large variety of problems, and are now widely used [39].

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behaviour, not something they struggle to learn!

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way [39].

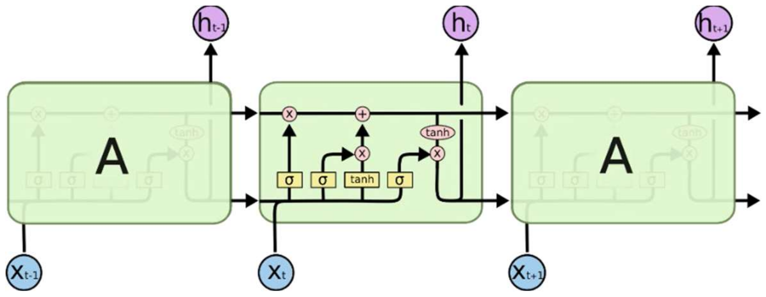


Fig 4.7: The repeating module in an LSTM contains four interacting layers [39]

Advantages of LSTM	Disadvantages of LSTM
- LSTM is great tool for anything that has a sequence. Since the meaning of a word depends on the ones that preceded it. This paved the way for NLP and narrative analysis to leverage Neural Networks. - LSTM can be used for text generation. You can train the model on the text of a writer, say, and the model will be able to generate new sentences that mimics the style the writer.	o They require a lot of resources and time to get trained and become ready for real-world applications. o LSTMs get affected by different random weight initializations and hence behave quite similar to that of a feed-forward neural net. They prefer small weight initializations instead. o LSTMs are prone to overfitting and it is difficult to apply the dropout algorithm to curb this issue. Dropout is a regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network.

Table 4.4: Advantages and disadvantages of LSTM [40]

Conclusion

In this chapter we present Deep learning concept and Deep neural network architectures. The next chapter will introduce the design of the system, dataset and the proposed architectures will be mentioned as well.

CHAPTER 5: DESING OF SYSTEM

Introduction

In practice, identifying the exact type of data drift is important, but identifying the impact on model performance and catching the drift on time so that actions such as retraining the model can be taken early on is more important.

In this chapter we present the global and detailed design of our system to solve the drift problem.

1. Methodology

Generally, our drift detection system will be following certain steps. Fig 4.1 represents the designed system.

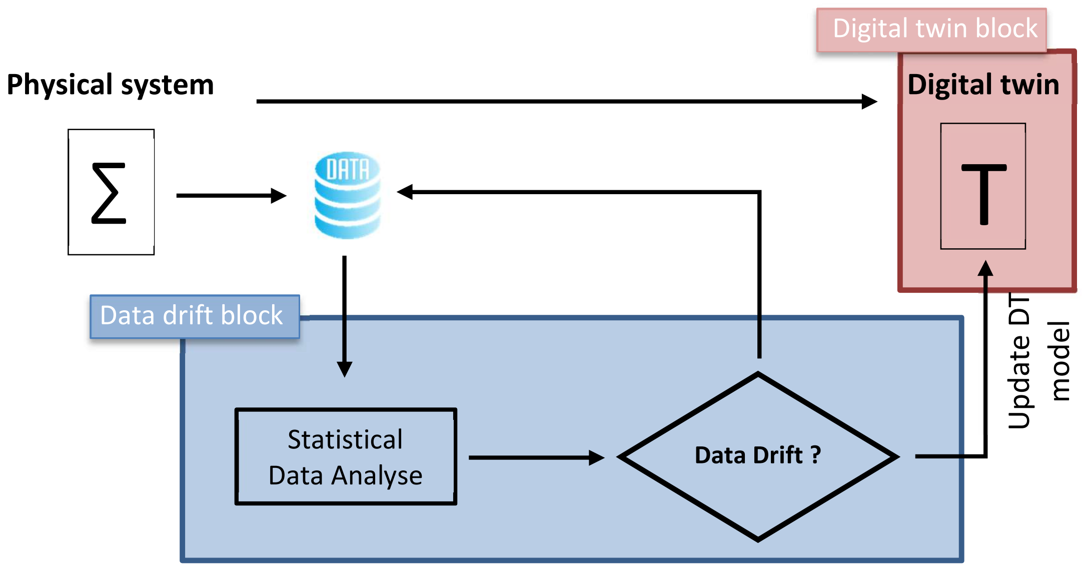


Fig 5.1: General Block diagram of the proposed system

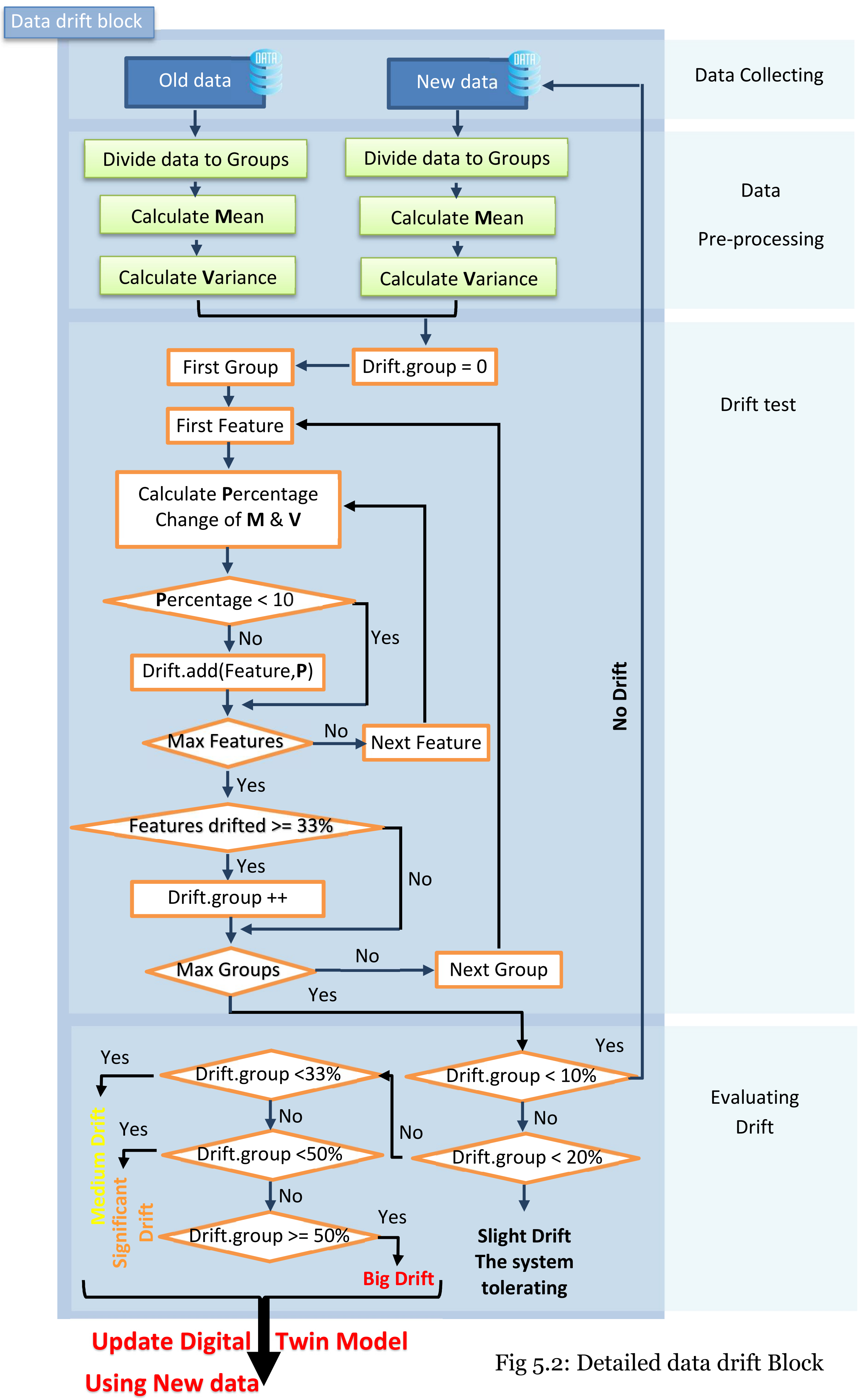


Fig 5.2: Detailed data drift Block

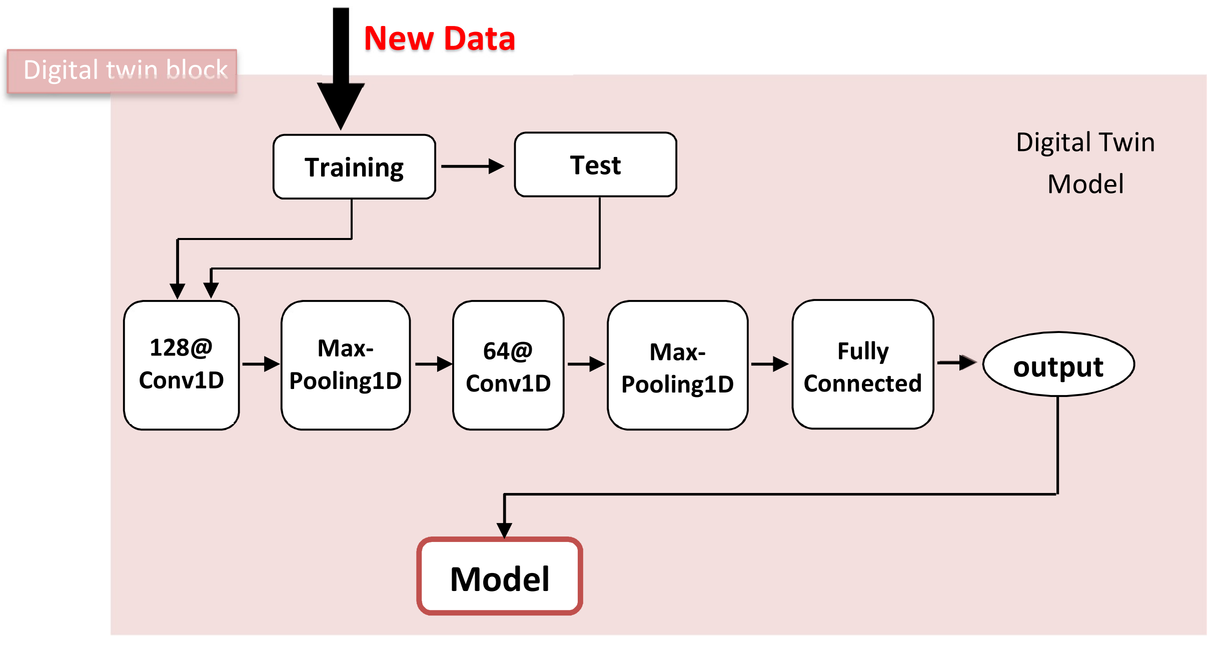


Fig 5.3: Detailed digital twin CNN model Block

The system starts by getting the data from the dataset. In our proposal, we are going to be using Instagram Concept Drifted Datasets. In next step, a pre-processing will be applicate on the data and the dataset structure.

In the third phase, we feed our Statistical Data Analyse System with the data to learn and detect if is there a data drift or not.

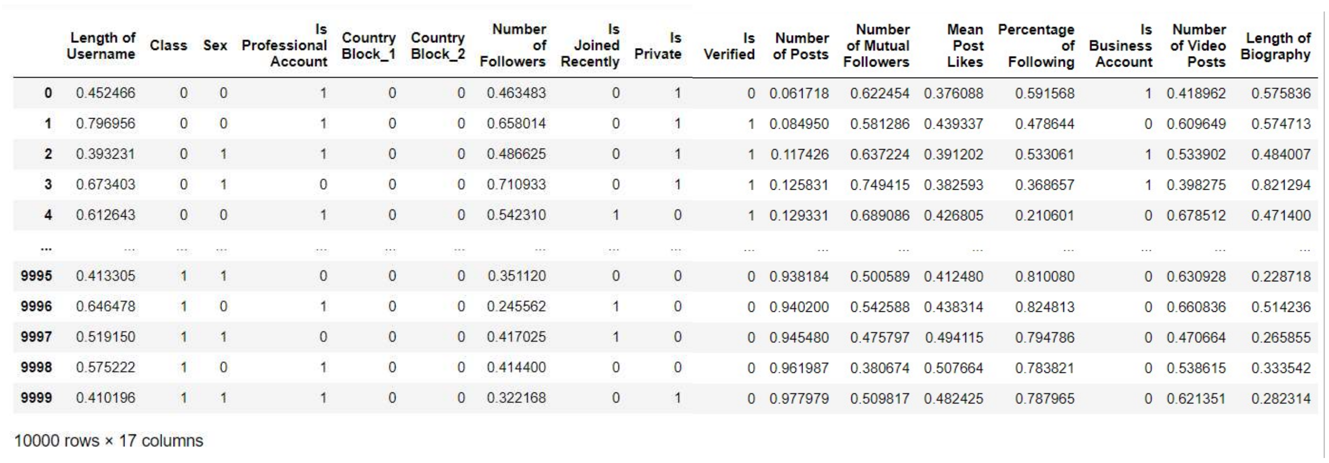
a. Dataset description

The dataset will have 16 features and one target. Note the 16 features was places in least correlated to most correlated to the target variable order, based on the collected dataset. Also note that originally the data collected had more features but any features with an absolute correlation value lower that 0.1 to the target variable was eliminated. Additionally, we have a binary class target variable that indicates if the user account follows the business account, denoted by a 1, or if the user account does not follow the business account, denoted by a 0 (i.e. 5000 user accounts of each class type).

○ File structure

The dataset will be composed of the following 16 features [42]:

- Is Business Account: a boolean feature on whether the account is a business account or not
- Country Block_1: a boolean feature that describes if the user is from the second most popular country the majority of the testing business's account's followers are from



	Length of Username	Class	Sex	Is Professional Account	Country Block_1	Country Block_2	Number of Followers	Is Joined Recently	Is Private	Is Verified	Number of Posts	Number of Mutual Followers	Mean Post Likes	Percentage of Following	Is Business Account	Number of Video Posts	Length of Biography
0	0.452466	0	0	1	0	0	0.463483	0	1	0	0.061718	0.622454	0.376088	0.591568	1	0.418962	0.575836
1	0.796956	0	0	1	0	0	0.658014	0	1	1	0.084950	0.581286	0.439337	0.478644	0	0.609649	0.574713
2	0.393231	0	1	1	0	0	0.486625	0	1	1	0.117426	0.637224	0.391202	0.533061	1	0.533902	0.484007
3	0.673403	0	1	0	0	0	0.710933	0	1	1	0.125831	0.749415	0.382593	0.368657	1	0.398275	0.821294
4	0.612643	0	0	1	0	0	0.542310	1	0	1	0.129331	0.689086	0.426805	0.210601	0	0.678512	0.471400
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	0.413305	1	1	0	0	0	0.351120	0	0	0	0.938184	0.500589	0.412480	0.810080	0	0.630928	0.228718
9996	0.646478	1	0	1	0	0	0.245562	1	0	0	0.940200	0.542588	0.438314	0.824813	0	0.660836	0.514236
9997	0.519150	1	1	0	0	0	0.417025	1	0	0	0.945480	0.475797	0.494115	0.794786	0	0.470664	0.265855
9998	0.575222	1	0	1	0	0	0.414400	0	0	0	0.961987	0.380674	0.507664	0.783821	0	0.538615	0.333542
9999	0.410196	1	1	1	0	0	0.322168	0	1	0	0.977979	0.509817	0.482425	0.787965	0	0.621351	0.282314

10000 rows × 17 columns

Fig 5.4: Dataset structure

- Length of Username: a numerical feature of the length of the username: (Note we choose this seemingly arbitrary feature since some engagement strategies suggest including the descriptive niche in the username to improve search results)
- Country Block_2: a boolean feature of whether the user is from the same country as the testing business account in question
- Is Professional Accoun: a boolean feature of whether the account is a professional account or not: (Note a business account and a professional account are two separate modes on the Instagram platform.)
- Number of Video Posts: a numerical feature that describes the number of posted videos aka InstaReels
- Sex: a boolean feature that describes gender of the user
- Is Joined Recently: a boolean feature that describes if the user has recently joined Instagram
- Is Verified: a boolean feature that describes if the user's identity is verified or not by Instagram
- Is Private: a boolean feature that describes whether the user's account is private or not
- Length of Biography: a numerical feature that describes the length of the user's biography
- Mean Post Likes: a numerical feature that describes the mean number of likes the user gets on their posts
- Number of Mutual Followers: a numerical feature that describes the number of mutual followers between the testing business account and the user's account
- Number of Followers: a numerical feature that describes the user's number of followers
- Number of Posts: a numerical feature that describes the user's number of posts

- Percentage of Following: a numerical feature that describes the percentage of the user's followers that the user itself follows

We must make a pre-processing and restructure the dataset to be able to use it in our detection drift system.

b. Pre-processing

In this step we will in first eliminate all Boolean features, then we will divide the dataset to groups, then we will flow the following steps:

- Calculate the mean for each feature

$$mean = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Calculate the variance for each feature

$$variance = S^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

c. Test the system

Firstly, in this step, we will calculate the Percentage Change in Mean & Variance for each feature between the historical data (old data) and the online data (new data).

$$Percentage\ Change = \frac{Final\ Value - Initial\ Value}{|Initial\ Value|}\ X\ 100$$

Then, we have three possible situations:

- if the Percentage change < 10 ➔ there is no drift in data
- 10 >= if the Percentage change < 20 ➔ that's mean that a slight drift is detected in data.
- if the Percentage change < 33 ➔ that's mean that a medium drift is detected in data ➔ a change is required to the model
- if the Percentage change < 50 ➔ that's mean that a Significant drift is detected in data ➔ Significant change is required to the model
- if the Percentage change >= 50 ➔ that's mean that a big drift is detected in data ➔ update required to the model

d. Digital twin model

In order to implement our model, we split the dataset into two tasks: training and testing.

Training and testing:

The process of training a model involves providing a deep learning algorithm (in our case CNN) with training data to learn from. The term CNN model refers to the model artifact that is created by the training process. We use the CNN model to get classification on new data.

Training set
Test set

Fig 5.5: Partition of Dataset

By the end of the learning phase the learning phase on the dataset, we save our model.

Model parameters:

- ✓ The learning rate: Is introduced as a constant (usually very small), in order to force the weight to get updated very smoothly and slowly (to avoid big steps and chaotic behaviour).
- ✓ Epoch: One epoch is when the entire dataset is passed forward and backward through the neural network only once. Epoch parameters refers to how many times the models aim to use the complete dataset for training.
- ✓ Batch: While training, the dataset is divided to a number of batches/parts, bigmini-batches are more computationally efficient.
- ✓ Batch size: Total number of training examples present in a single batch.
- ✓ Loss functions: Is a performance metric on how well the neural net manages to reach its goal of generating outputs as close as possible to the desired values loss.

Model evaluating:

Accuracy measures how well our model predicts by comparing the model predictions with the true values in terms of percentage [56].

Loss is a value that represents the summation of errors in our model. It measures how well (or bad) our model is doing. If the errors are high, the loss will be high, which means that the model does not do a good job. Otherwise, the lower it is, the better our model works [56].

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions}$$

Or

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

$$loss = \sum (Desiredoutput - Actualoutput\ )^2$$

	Low Loss	**High Loss**
Low Accuracy	A lot of small errors	A lot of big errors
High Accuracy	A few small errors	A few big errors

Table 5.1: Accuracy and loss table

Conclusion

The objective of this chapter was to present the general and detailed design of our system for drift detection using a statistical data analyse.

According to the dataset and our system architectures we used, we could deduce in the next chapter how to implement our model and discuss the experiments and results.

CHAPTER 6: IMPLEMENTATION AND RESULTS

Introduction

After having detailed the phases of our system in the previous chapter, in this chapter we present the steps proposed to achieve the system designed.

We start in the first part, by specifying the frameworks and tools used in development. Eventually, we are going to explain all the experiments that we have applied and discuss the results obtained. Finally, we are interested in presenting some parts of the developed code.

1. Implementation frameworks and tools

Deep learning frameworks offer building blocks for designing, training, and validating deep neural networks through a high-level programming interface [43]. Deep learning frameworks can help you upload data and train a deep learning model that would lead to accurate and intuitive predictive analysis. Without these tools it would be very hard for scientists to work on deep learning tasks.

In this section we review the different tools and frameworks used in this thesis.

1.1. Google colab

Colaboratory, or Colab for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs [44].

1.2. JupyterLab

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality [45].

1.3. Anaconda

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today [46].

1.4. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed [47].

1.5. TensorFlow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow was developed by the Google Brain team for internal Google use in research and production. TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as JavaScript, C++, and Java. This flexibility lends itself to a range of applications in many different sectors [48].

1.6. Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. Keras is Simple, Flexible and Powerful [49].

1.7. NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more [50].

1.8. Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. It can be combined with the scientific computing python libraries NumPy and SciPy [51].

1.9. Scikit-learn

Scikit-learn and also known as sklearn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy [52].

1.10. Pandas

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language [53].

2. Implementation phases

As we see in the global system design, the implementation has several phases.

2.1. Dataset Description

The characteristics of the dataset used are summarized in the table below

Number of Features	**16**
Number of simples	**10000**
Format	**CSV File**

Table 6.1: Dataset description.

The dataset consists of a set of approximately 10,000 rows and 17 columns (16 features and 1 target)

2.2. Loading and pre-processing the dataset

2.3.1. Import Libraries

First, we start by importing the different libraries and modules needed to work with the data as presented below.

```python
import numpy as np
import cv2
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
import pandas as pd
import seaborn as sn
from keras.callbacks import EarlyStopping
from tensorflow.keras.utils import to_categorical
```

Listing 6.1: Libraries used

2.3.2. Import Dataset

We begin with importing the dataset using read_csv(file_name) function from panda library since the dataset used is CSV format.

```python
Initial_Data = pd.read_csv('training_data.csv')
```

Listing 6.2: Import dataset

2.3.3. Visualise data

Flowing is the code and the result to visualise the data

```python
def Visualise_data(Data,features, is_boolian, name_file):
    if is_boolian == 1:
        kwargs = dict(alpha=1, bins=2)
    else:
        kwargs = dict(alpha=1, bins=10)
    fig = plt.figure(figsize=(40, 30))
    for i in range(len(features)):
        ax=plt.subplot(3,3,i+1)
        plt.hist(Data[features[i]], **kwargs, color="#9ceabc", label=features[i], rwidth=0.98)
        ax.set_xlabel(features[i], labelpad=15, fontsize=12, color="#333533");
        if is_boolian == 1:
            plt.xticks([0,1])
    plt.savefig(name_file)
    plt.show()
```

Listing 6.3: Data visualisation procedure

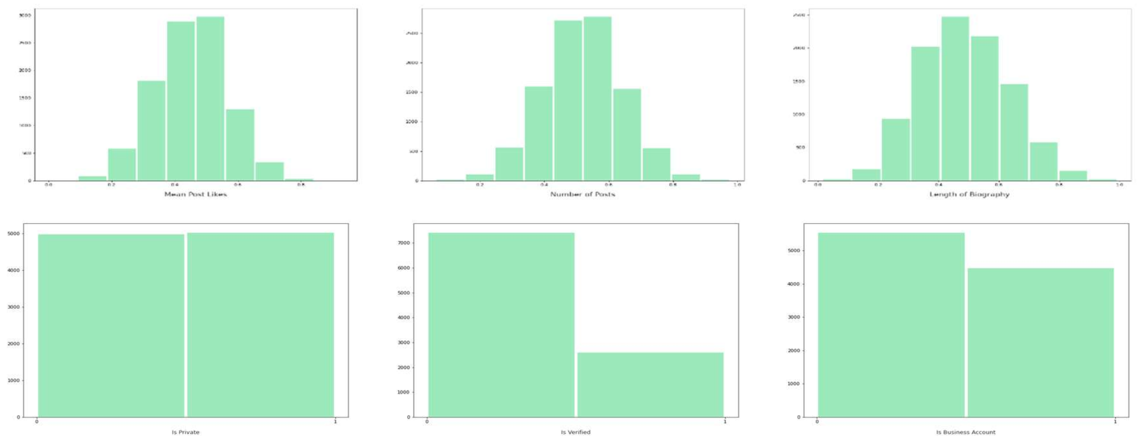
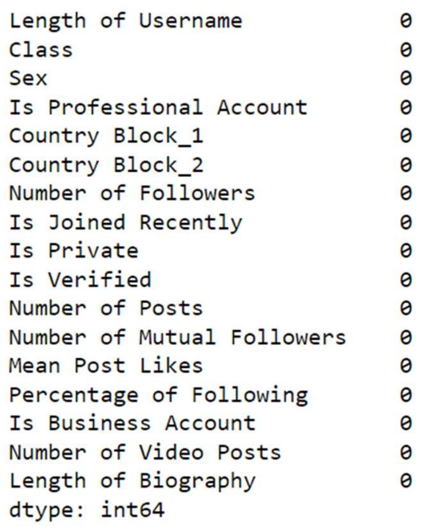


Fig 6.1: Examples of data visualisation

2.3.4. Checking for missing values

Checking for missing values is an important step because it influences the performance of our models, we used the isna().sum() functions to perform this step.



```
Length of Username           0
Class                        0
Sex                          0
Is Professional Account      0
Country Block_1              0
Country Block_2              0
Number of Followers          0
Is Joined Recently           0
Is Private                   0
Is Verified                  0
Number of Posts              0
Number of Mutual Followers   0
Mean Post Likes              0
Percentage of Following      0
Is Business Account          0
Number of Video Posts        0
Length of Biography          0
dtype: int64
```

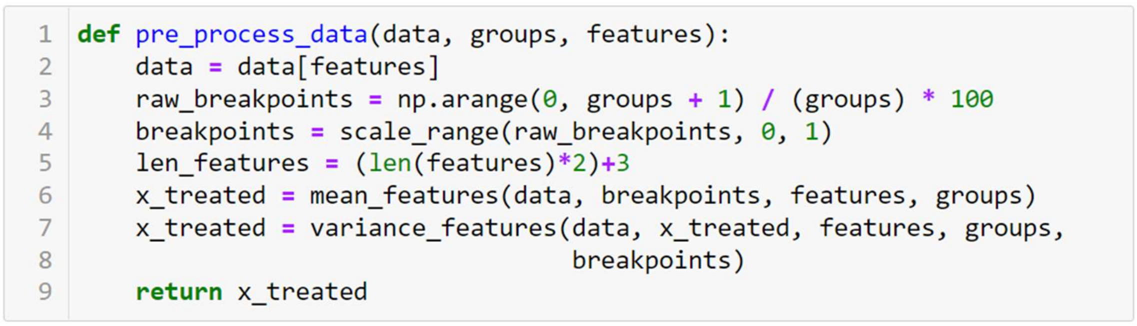
Fig 6.2: Dataset Verification for missing values

2.3. Test the system

Now we eliminate unused features for detecting the data drift then we calculate the mean, Variance of each remaining feature after we divide the data into groups. We do this calculation for the old data (training_data.csv) and the new data (data.csv). Next we calculate the percentage of change of mean and variance, between the old and the new data, for each feature.

2.3.1. Calculating Mean and variance

We calculate Mean and the variance in each group for each feature to the old and new data as a first step.



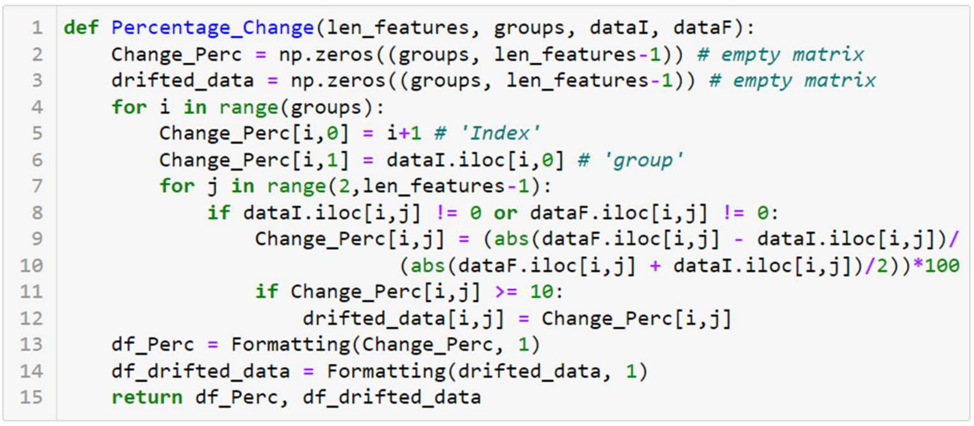
```python
def pre_process_data(data, groups, features):
    data = data[features]
    raw_breakpoints = np.arange(0, groups + 1) / (groups) * 100
    breakpoints = scale_range(raw_breakpoints, 0, 1)
    len_features = (len(features)*2)+3
    x_treated = mean_features(data, breakpoints, features, groups)
    x_treated = variance_features(data, x_treated, features, groups,
                                  breakpoints)
    return x_treated
```

Listing 6.4: Pre-Process data

2.3.2. Calculating Percentage of change

We calculate the percentage of change in each group for each feature between the old and new data. A drifted feature is a feature that the percentage of change exceed 0.1.

- If we find less than 10% of groups (in each group 33% of features are drifted) ➔ no significant drift detected
- If we find that up to 10% and less than 20% of groups (in each group 33% of features are drifted) ➔ we have a slight drift.
- If we find that up to 20% and less than 33% of groups (in each group 33% of features are drifted) ➔ we have a medium drift ➔ Update Digital twin model
- If we find that up to 33% and less than 50% of groups (in each group 33% of features are drifted) ➔ we have a significant drift ➔ Update Digital twin model
- If we find that up to 50% of groups (in each group 33% of features are drifted) ➔ we have a big drift ➔ Update Digital twin model



```python
def Percentage_Change(len_features, groups, dataI, dataF):
    Change_Perc = np.zeros((groups, len_features-1)) # empty matrix
    drifted_data = np.zeros((groups, len_features-1)) # empty matrix
    for i in range(groups):
        Change_Perc[i,0] = i+1 # 'Index'
        Change_Perc[i,1] = dataI.iloc[i,0] # 'group'
        for j in range(2,len_features-1):
            if dataI.iloc[i,j] != 0 or dataF.iloc[i,j] != 0:
                Change_Perc[i,j] = (abs(dataF.iloc[i,j] - dataI.iloc[i,j])/
                            (abs(dataF.iloc[i,j] + dataI.iloc[i,j])/2))*100
                if Change_Perc[i,j] >= 10:
                    drifted_data[i,j] = Change_Perc[i,j]
    df_Perc = Formatting(Change_Perc, 1)
    df_drifted_data = Formatting(drifted_data, 1)
    return df_Perc, df_drifted_data
```

Listing 6.5: Percentage change calculation

2.3.3. Updating digital twin model

We use a convolutional neural network as deep learning model to create a digital twin model for classification. We use as accuracy and loss as evaluation metrics.

Eposh = 50

BatchSize = 100

Total rows = 10000

```python
def update_model(New_Data, groups):
    #load data
    x = New_Data.drop('Class', axis=1)
    y = to_categorical(New_Data.loc[:,'Class'])
    x_train, x_val, y_train, y_val = train_test_split(x, y, test_size = 0.2)
    #create model
    model = keras.Sequential()
    #add layers to model
    model.add(keras.layers.Conv1D(128,3,activation='relu', padding='same',input_shape=(x.shape[1],1)))
    model.add(keras.layers.MaxPooling1D(2, padding='same'))
    model.add(keras.layers.Conv1D(64,3,activation='relu', padding='same'))
    model.add(keras.layers.MaxPooling1D(2, padding='same'))
    model.add(keras.layers.Flatten())
    model.add(keras.layers.Dense(32, activation='relu'))
    model.add(keras.layers.Dense(2))
    print(model.summary())
    #State optimizer, loss and evaluation metric
    model.compile(optimizer='adam', loss='categorical_crossentropy',  metrics=['accuracy'])
    #Train the model
    Epochs = 50
    batchSize = 100
    history = model.fit(x_train, y_train, validation_data=(x_val, y_val),epochs=Epochs, batch_size=batchSize)
    #list all data in history
    print(history.history.keys())
    #plot accuracy and loss
    plot_accuracy_result(history)
    plot_loss_result(history)
```

Listing 6.6: Update Digital twin model function

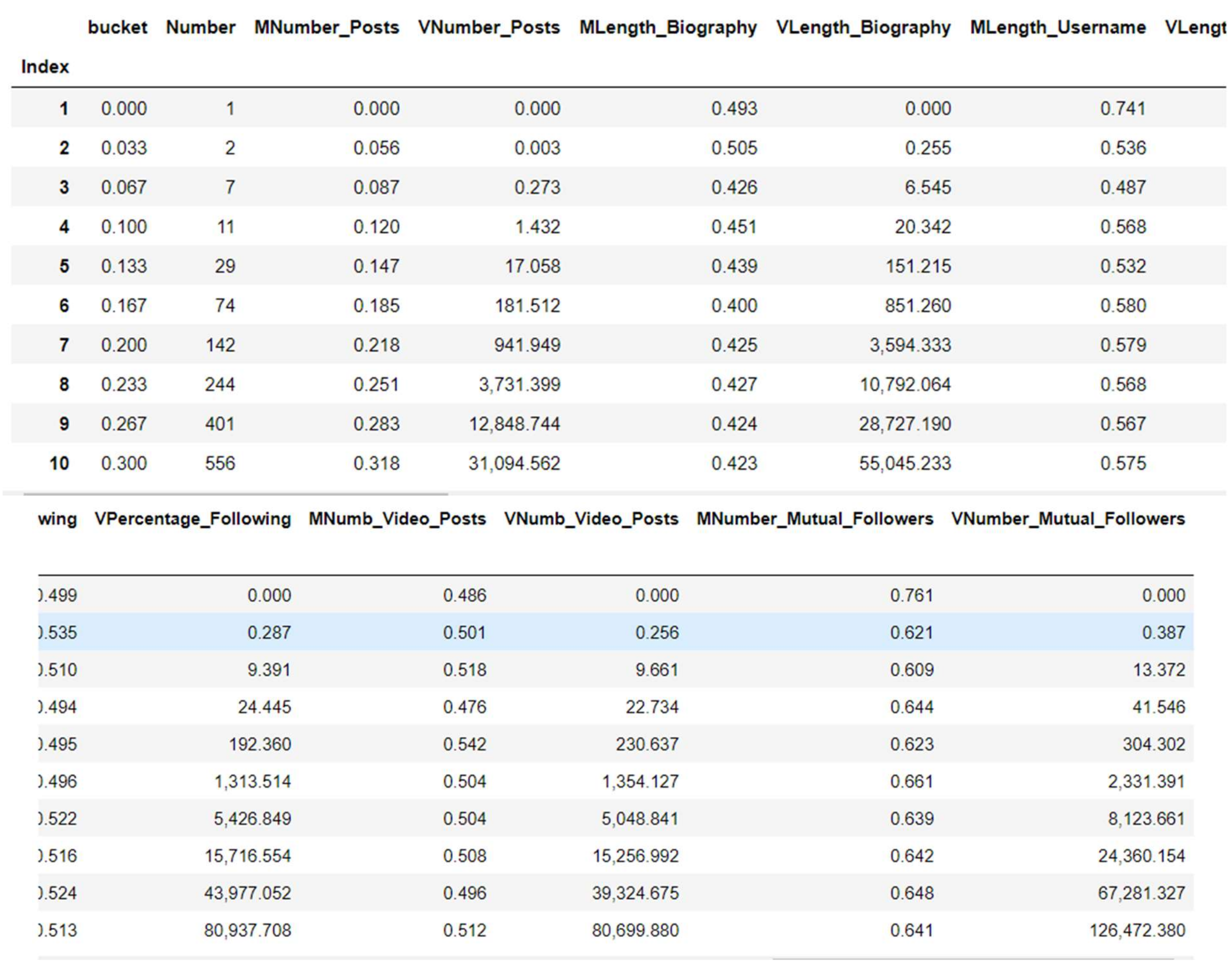
3. Experiments and results

This section is the essence of our work. our experiments were conducted on our dataset, and we analysed the obtained result.

In this experiment we load, check for missing values then pre-process data (old, new). Eliminate unused features the we divide the data to groups (50 groups).

```python
prep_Data = pre_process_data(Initial_Data, groups, features)
data = Formatting(prep_Data, 0)
data.head(10)
```

Listing 6.7: Pre-process data function call



Index	bucket	Number	MNumber_Posts	VNumber_Posts	MLength_Biography	VLength_Biography	MLength_Username	VLengt
1	0.000	1	0.000	0.000	0.493	0.000	0.741	
2	0.033	2	0.056	0.003	0.505	0.255	0.536	
3	0.067	7	0.087	0.273	0.426	6.545	0.487	
4	0.100	11	0.120	1.432	0.451	20.342	0.568	
5	0.133	29	0.147	17.058	0.439	151.215	0.532	
6	0.167	74	0.185	181.512	0.400	851.260	0.580	
7	0.200	142	0.218	941.949	0.425	3,594.333	0.579	
8	0.233	244	0.251	3,731.399	0.427	10,792.064	0.568	
9	0.267	401	0.283	12,848.744	0.424	28,727.190	0.567	
10	0.300	556	0.318	31,094.562	0.423	55,045.233	0.575	

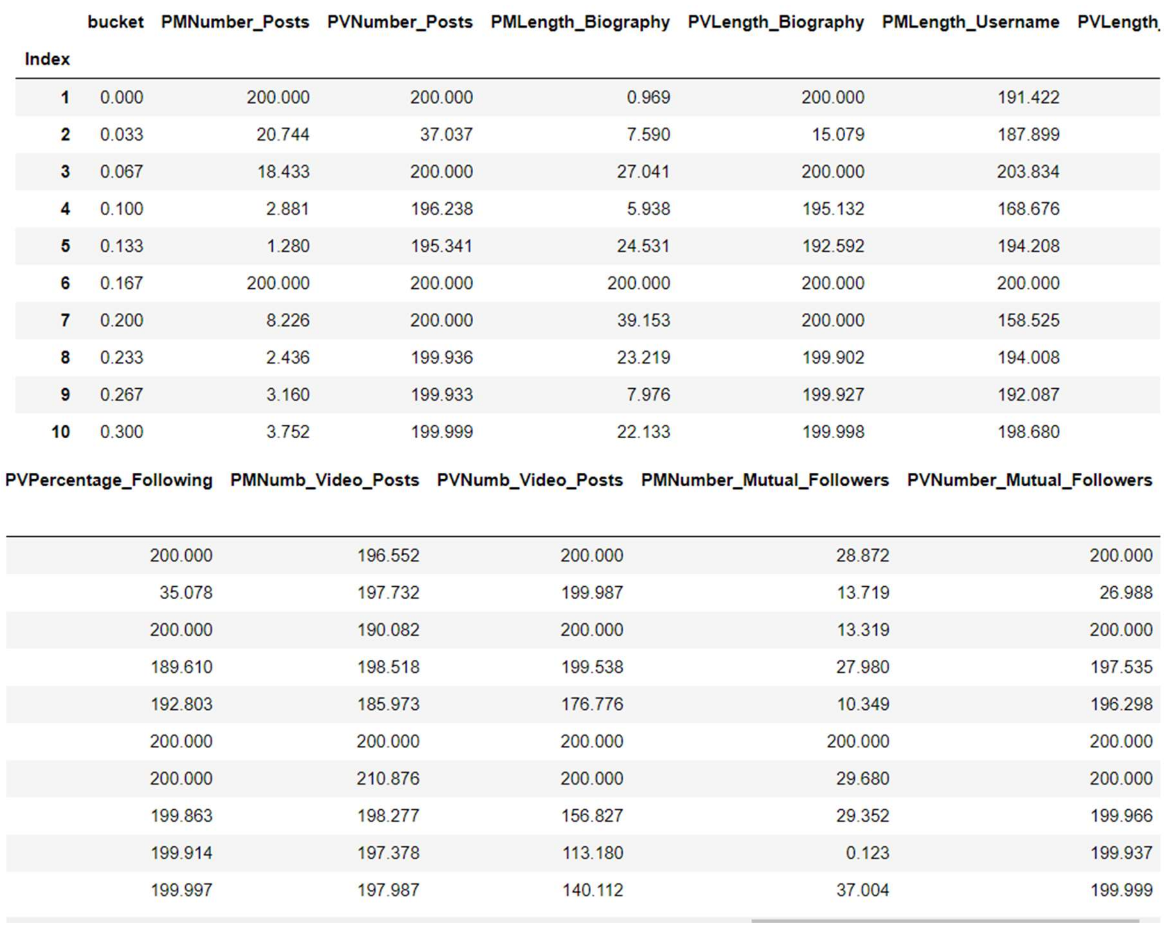
wing	VPercentage_Following	MNumb_Video_Posts	VNumb_Video_Posts	MNumber_Mutual_Followers	VNumber_Mutual_Followers
).499	0.000	0.486	0.000	0.761	0.000
).535	0.287	0.501	0.256	0.621	0.387
).510	9.391	0.518	9.661	0.609	13.372
).494	24.445	0.476	22.734	0.644	41.546
).495	192.360	0.542	230.637	0.623	304.302
).496	1,313.514	0.504	1,354.127	0.661	2,331.391
).522	5,426.849	0.504	5,048.841	0.639	8,123.661
).516	15,716.554	0.508	15,256.992	0.642	24,360.154
).524	43,977.052	0.496	39,324.675	0.648	67,281.327
).513	80,937.708	0.512	80,699.880	0.641	126,472.380

Fig 6.3: Result calculation of mean and variance

Next, we calculate the percentage of change in each group for each feature between the old and new data. The result is in the figure below:

```
Perc_Chan = Percentage_Change(len_features, groups, data, New_data)
Perc_Chan.head(10)
```

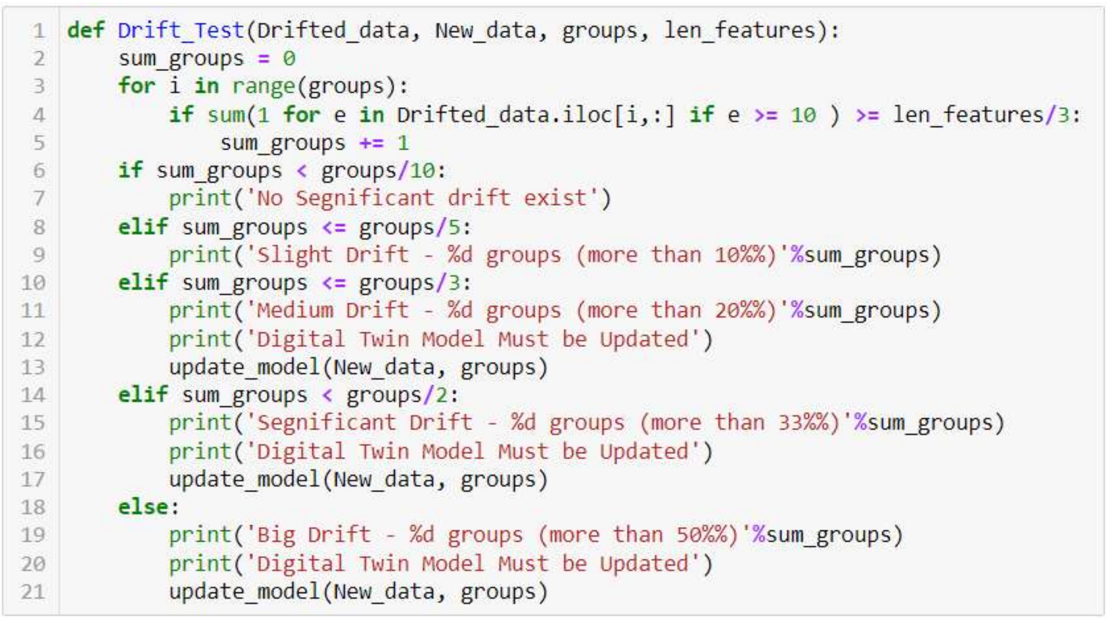
Listing 6.8: Percentage of change calculation call



Index	bucket	PMNumber_Posts	PVNumber_Posts	PMLength_Biography	PVLength_Biography	PMLength_Username	PVLength_	PVPercentage_Following	PMNumb_Video_Posts	PVNumb_Video_Posts	PMNumber_Mutual_Followers	PVNumber_Mutual_Followers
1	0.000	200.000	200.000	0.969	200.000	191.422		200.000	196.552	200.000	28.872	200.000
2	0.033	20.744	37.037	7.590	15.079	187.899		35.078	197.732	199.987	13.719	26.988
3	0.067	18.433	200.000	27.041	200.000	203.834		200.000	190.082	200.000	13.319	200.000
4	0.100	2.881	196.238	5.938	195.132	168.676		189.610	198.518	199.538	27.980	197.535
5	0.133	1.280	195.341	24.531	192.592	194.208		192.803	185.973	176.776	10.349	196.298
6	0.167	200.000	200.000	200.000	200.000	200.000		200.000	200.000	200.000	200.000	200.000
7	0.200	8.226	200.000	39.153	200.000	158.525		200.000	210.876	200.000	29.680	200.000
8	0.233	2.436	199.936	23.219	199.902	194.008		199.863	198.277	156.827	29.352	199.966
9	0.267	3.160	199.933	7.976	199.927	192.087		199.914	197.378	113.180	0.123	199.937
10	0.300	3.752	199.999	22.133	199.998	198.680		199.997	197.987	140.112	37.004	199.999

Fig 6.4: Result calculation of Percentage of change

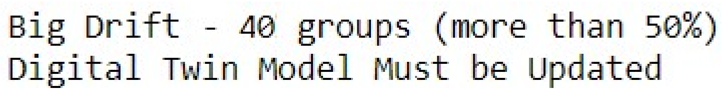
Finally, we test the results calculation using Drift_Test() function.



```python
def Drift_Test(Drifted_data, New_data, groups, len_features):
    sum_groups = 0
    for i in range(groups):
        if sum(1 for e in Drifted_data.iloc[i,:] if e >= 10 ) >= len_features/3:
            sum_groups += 1
    if sum_groups < groups/10:
        print('No Segnificant drift exist')
    elif sum_groups <= groups/5:
        print('Slight Drift - %d groups (more than 10%%)'%sum_groups)
    elif sum_groups <= groups/3:
        print('Medium Drift - %d groups (more than 20%%)'%sum_groups)
        print('Digital Twin Model Must be Updated')
        update_model(New_data, groups)
    elif sum_groups < groups/2:
        print('Segnificant Drift - %d groups (more than 33%%)'%sum_groups)
        print('Digital Twin Model Must be Updated')
        update_model(New_data, groups)
    else:
        print('Big Drift - %d groups (more than 50%%)'%sum_groups)
        print('Digital Twin Model Must be Updated')
        update_model(New_data, groups)
```

Fig 6.9: Drift test function

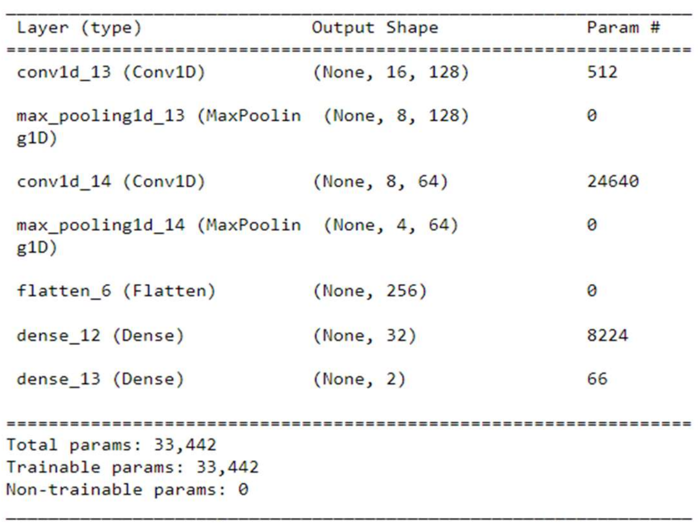
In our data tested, we find that the percentage of change in more than 50% of groups, more than 30% of features are drifted and this mean that there is a big data drift and we must update our digital twin model using new data to create new model.



```
Big Drift - 40 groups (more than 50%)
Digital Twin Model Must be Updated
```

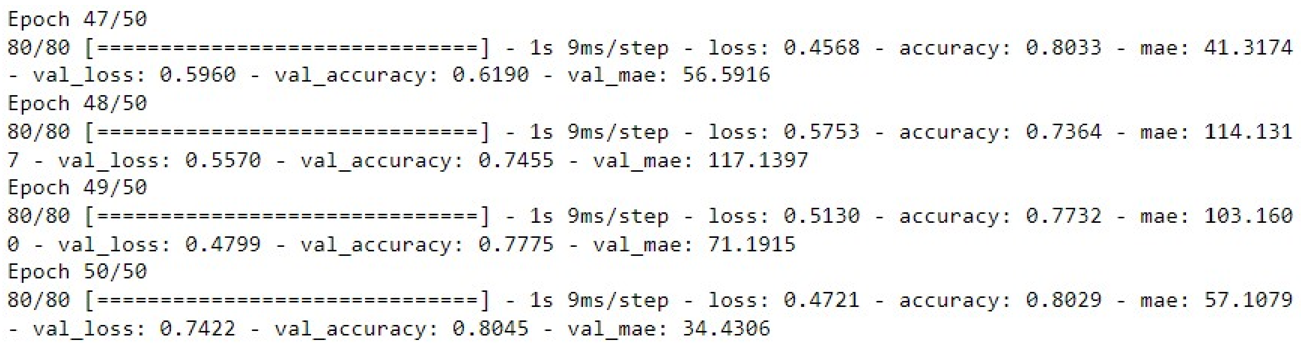
Fig 6.5: Result of Drift test

According to the result of drift test, the system retrains the CNN to get the new model. figures below show the results after retraining the model.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_13 (Conv1D)          (None, 16, 128)           512

 max_pooling1d_13 (MaxPoolin  (None, 8, 128)           0
 g1D)

 conv1d_14 (Conv1D)          (None, 8, 64)             24640

 max_pooling1d_14 (MaxPoolin  (None, 4, 64)            0
 g1D)

 flatten_6 (Flatten)         (None, 256)               0

 dense_12 (Dense)            (None, 32)                8224

 dense_13 (Dense)            (None, 2)                 66

=================================================================
Total params: 33,442
Trainable params: 33,442
Non-trainable params: 0
_________________________________________________________________
```

Fig 6.6: Summary CNN model used



```
Epoch 47/50
80/80 [==============================] - 1s 9ms/step - loss: 0.4568 - accuracy: 0.8033 - mae: 41.3174
- val_loss: 0.5960 - val_accuracy: 0.6190 - val_mae: 56.5916
Epoch 48/50
80/80 [==============================] - 1s 9ms/step - loss: 0.5753 - accuracy: 0.7364 - mae: 114.131
7 - val_loss: 0.5570 - val_accuracy: 0.7455 - val_mae: 117.1397
Epoch 49/50
80/80 [==============================] - 1s 9ms/step - loss: 0.5130 - accuracy: 0.7732 - mae: 103.160
0 - val_loss: 0.4799 - val_accuracy: 0.7775 - val_mae: 71.1915
Epoch 50/50
80/80 [==============================] - 1s 9ms/step - loss: 0.4721 - accuracy: 0.8029 - mae: 57.1079
- val_loss: 0.7422 - val_accuracy: 0.8045 - val_mae: 34.4306
```

Fig 6.7: Execution of training model result

Conclusion

In this chapter, we have described the detailed design for our system with the tools used for developing of this system. Then, we also showed some portions of our code and discussed the obtained results which were later illustrated in figures.

GENERAL CONCLUSION

General Conclusion

This work aims to develop and apply a system model to detect drift in data and update the digital twin to make it accurate. To achieve research objectives, we conducted experiments using the designed system on Instagram Concept Drifted Datasets dataset.

The results have shown that the statistical system detection designed can give very interesting results for data drift detection and can offers a new, simple and effective method for drift detection.

Our approach shows good results in terms performance, but we must teste our system detection on different datasets to see if they can perform as well as they did in this dataset.

Limitation

After the quality of the results obtained but the manual choice of parameters (drift rate, number of groups) of the proposed system present a significant limit in the generation of the adequate model.

Perspectives

For the future, we would study further many related problems and test our system in different datasets. We also aim to integrate our system detection directly into a real environment to evaluate the result in the real world.

REFERRENCES

References

[1] "Digital Twin Definition & Value", An AIAA and AIA Position Paper, (December 2020).

[2] Aidan Fuller, Zhong Fan, Charles Day, Chris Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research", IEEE, 2020.

[3] Zongyan Wang, "Digital Twin Technology", 2020.

[4] "Digital Twins Creating Digital Operations Today to Deliver Business Value Tomorrow", Altran,2021.

[5] J. Trauer, S. Schweigert-Recksiek, C. Engel,K. Spreitzer, M. Zimmermann,"What Is a Digital Twin - Definitions and Insights from an Industrial Case Study in Technical Product Development", International Design Conference, 2020.

[6] Shoumen Palit Austin Datta, "Emergence of Digital Twins Is this the march of reason? ", 2016.

[7] Michael Grieves, "Digital Twin Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems", (August 2017).

[8] "Industry 4.0 and the digital twin", Deloitte University Press, 2017.

[9] Maggie Mae Armstrong, "Cheat sheet - What is Digital Twin", 2020.

[10] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, Wilfried Sihn,"Digital Twin in manufacturing A categorical literature review and classification", International Federation of Automatic Control PapersOnLine, 2018.

[11] Sonal, S. Reddy, and D. Kumar, "Review of Smart Health Monitoring Approaches with Survey Analysis and Proposed Framework", IEEE Internet of Things Journal, pp. 1–1, 2018.

[12] David Cearley et al, "Top 10 Strategic Technology Trends for 2018", (03-10-2017) [Online]. (Available: https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018)

[13] Yoo Youngjin, Boland Richard, Lyytinen Kalle, Majchrzak Ann, "Organizing for innovation in the digitized world", Organization Science. 23 (5): 1398–1408, (September–October 2012).

[14] Rishabh Bhatia, "Data Drift: An In-Depth Understanding", Data Scientist reporter at Predactica, (03/03/2022). [Online]. (Available: https://www.linkedin.com/pulse/data-drift-in-depth-understanding-rishabh-bhatia)

[15] "What Is Data Drift? How Smart Data Pipelines Help", StreamSets, 2022. [Online]. (Available: https://streamsets.com/why-dataops/what-is-data-drift/)

[16] Michał Oleszak, "Don't let your model's quality drift away", (20/07/2021). [Online]. (Available: https://towardsdatascience.com/dont-let-your-model-s-quality-drift-away-53d2f7899c09)

[17] "Data Drift and Machine Learning Model Sustainability", Analytics Insight (29/10/2020). [Online]. (Available: https://www.analyticsinsight.net/data-drift-and-machine-learning-model-sustainability)

[18] Prarthana Saikia, "The Importance of Data Drift Detection that Data Scientists Do Not Know", (15/10/2021). [Online]. (Available: https://www.analyticsvidhya.com/blog/2021/10/mlops-and-the-importance-of-data-drift-detection/)

[19] "Detect data drift (preview) on datasets", Microsoft Article, (11/11/2021). [Online]. (Available: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-monitor-datasets?tabs=python)

[20] Numal Jayawardena, "Data Drift - Types of Data Drift", (27/06/2021). [Online]. (Available: https://towardsdatascience.com/data-drift-part-1-types-of-data-drift-16b3eb175006)

[21] Jian-Wei Liao, Bi-Ru Dai ,"An Ensemble Learning Approach for Concept Drift", IEEE, 2014.

[22] "Statistical Analysis of Data for Data Scientists", (30/03/2021) [Online]. (Available: https://www.analyticsvidhya.com/blog/2021/03/statistical-analysis-of-data/)

[23] "What Is Statistical Analysis? - Chad Brooks", Business News Daily Staff, (19/10/2020) [Online]. (Available: https://www.businessnewsdaily.com/6000-statistical-analysis.html)

[24] Simona Micevska, "A Statistical Drift Detection Method", Master's Thesis, 2019.

[25] "Leared Statistics", (2020) [Online]. (Available: https://statistics.laerd.com/statistical-guides/descriptive-inferential-statistics-faqs.php)

[26] Priya Pedamkar, "Statistical Analysis in Python", EDUCBA, (2022). [Online]. (Available: https://www.educba.com/statistical-analysis-in-python/)

[27] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", O'Reilly Media, Inc., 2019.

[28] Tiar Ilyes, "A deep learning approach for intrusion detection in Vehicular Networks", Master dissertation, University of Biskra, 2021.

[29] S. Dargan, M. Kumar, M. R. Ayyagari and G. Kumar , "A Survey of Deep Learning and Its Applications: A New Paradigm", *Archives of Computational Methods in Engineering,* vol. 27, 2019.

[30] S.-C. Wang, "Artificial Neural Network," in *Interdisciplinary Computing in Java Programming*, Springer, Boston, MA, pp. 81-100, 2003.

[31] A. Krenker, J. Bešter and A. Kos, "Introduction to the Artificial Neural Networks", 2011.

[32] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning", 2018.

[33] A. Varangaonkar, "Top 5 Deep Learning Architectures", (24/07/2018). [Online]. (Available: https://hub.packtpub.com/top-5-deep-learning-architectures/).

[34] A. Voulodimos, N. Doulamis, A. Doulamis and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational intelligence and neuroscience,* 2018.

[35] "Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning," *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018), MAY 18th & 19th 2018,* 2018.

[36] P.-R. Chen, H.-M. Hang, S.-W. Chan and L. J. Jhih, "DSNet: An Efficient CNN for Road Scene Segmentation," *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE,* pp. 424-432, 2019.

[37] D. Mandic and J. A. Chambers, Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability, Wiley, 2001.

[38] H. Xu, H. Lu, G. Yang and C. Zhang, "Sentiment Analysis of Chinese Version Using SVM & RNN," *ICIE '17: Proceedings of the 6th International Conference on Information Engineering,* pp. 1-5, 2017.

[39] "Understanding LSTM Networks," 27 August 2015. [Online]. (Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/).

[40] "Understanding of LSTM Networks," [Online]. (Available: https://www.geeksforgeeks.org/understanding-of-lstm-networks/).

[41] Tan, M., & Le, Q. v., "EfficientNet: Rethinking model scaling for convolutional neural networks". 36th International Conference on Machine Learning, ICML 2019.

[42] "Concept Drifted Data - Instagram Concept Drifted Datasets". [Online]. (Available: https://www.kaggle.com/datasets/gabriellacolletti/concept-drifted-data)

[43] "Deep Learning Frameworks", [Online]. (Available: https://developer.nvidia.com/deep-learning-frameworks)

[44] "Google Colab", [Online]. (Available: https://research.google.com/colaboratory/faq.html)

[45] "JupyterLab", [Online]. (Available: https://jupyter.org/)

[46] "Anaconda" [Online]. (Available: https://www.anaconda.com/)

[47] "Python". [Online]. (Available: https://www.python.org/doc/essays/blurb/)

[48] "Tensorflow", [Online]. (Available: https://en.wikipedia.org/wiki/TensorFlow)

[49] "Keras", [Online]. (Available: https://keras.io/about/)

[50] "What is NumPy",
[Online]. (Available : https://numpy.org/doc/stable/user/whatisnumpy.html)

[51] "Matplotlib: Visualization with Python", [Online]. (Available : https://matplotlib.org/)

[52] "Scikit-learn", [Online]. (Available: https://en.wikipedia.org/wiki/Scikit-learn)

[53] "Pandas", [Online]. (Available: https://pandas.pydata.org/)

[54] "Univariate, Bivariate and Multivariate data and its analysis", [Online]. (Available: https://www.geeksforgeeks.org/univariate-bivariate-and-multivariate-data-and-its-analysis/)

[55] Rob Kabacoff, "Data Visualization with R", (01-12-2020).
[Online]. (Available: https://rkabacoff.github.io/datavis/Multivariate.html)

[56] Martin Riva, "Interpretation of loss and accuracy", (19-01-2021). [Online]. (Available: https://www.baeldung.com/cs/ml-loss-accuracy)

[57] Geoffrey I. Webb, Loong Kuan Lee, Bart Goethals, Franc¸ois Petitjean, "Analyzing concept drift and shift from sample data", Data Mining and Knowledge Discovery, pp. 1179-1199, 2018.