Summer Internship Project
Report

Topic

Anomaly Detection AI Service - Annotation

Submitted by

Rajneesh Pandey
106119100
National Institute of Technology, Tiruchirappalli

Under the guidance of

Satishkumar Venkatasamy
Senior Director, Oracle Cloud Infrastructure Artificial Intelligence Services

BU - Oracle Cloud Infrastructure
Oracle India Private Limited India Development Center
Oracle Technology Park 3, Bannerghatta Road, Bangalore - 560 029, India

Summer Internship 2022

Acknowledgment

Firstly, I would like to thank the Training and Placement Cell of National Institute of Technology, Tiruchirappalli for allowing me to sit in the campus internship process and also the faculty of the Computer Science and Engineering department whose valuable lessons helped me a lot during the entire process.

I would also like to express my gratitude to the Campus Recruitment team of Oracle for offering me an internship for a duration of 8 weeks and an opportunity to work with the Oracle Cloud Infrastructure (OCI) team, and also for carrying out the entire virtual internship process smoothly.

I would like to thank my mentor Ms. Ritanshu Jaiswal, my manager Mr. Satishkumar Venkatasamy and the entire Oracle OCI-AI Services team who motivated, helped and guided me throughout my internship project.

This opportunity is a big milestone in my life and career. I strive to use the skills and knowledge gained in the best possible way, and continue to work on the improvements suggested by my team, in order to attain desired career objectives.

Finally, I would like to acknowledge the incredible amount of efforts put by my parents to make this experience possible. Their constant support and encouragement have what made the internship a lifelong learning lesson.

Sincerely,
Rajneesh Pandey

BU - Oracle Cloud and Infrastructure

Oracle India Pvt. Ltd. India Development Center

Certificate with Signatures and Seal of the Industry Person

Satishkumar Venkatasamy
(Project Guide)

Date:

Abstract

The project is a **Anomaly Detection AI Service - Annotation**(full stack web Application) utilise as **Internal tool in Oracle Cloud Infrastructure(OCI) Artificial Intelligence (AI) service.**

Anomaly detection aims at finding unexpected or rare events in data streams, commonly referred to as anomalous events. Detecting anomalies could be useful directly or as a first insight to find new knowledge in the data.

Process of finding outliers in a given dataset. Outliers are the data objects that stand out amongst other objects in the dataset and do not conform to the normal behavior in a dataset.

It is a data science application that combines multiple data science tasks like classification, regression, and clustering. The target variable to be predicted is whether a transaction is an outlier or not. Since clustering tasks identify outliers as a cluster, distance-based and density-based clustering techniques can be used in anomaly detection tasks.

Anomaly detection is fundamental in many applications, most notably in real-time applications, where spotting anomalies is vital such as in *health*, *critical infrastructures*, *security applications*, to name a few.

The project will have a huge impact as it eases the tiring process of Manually detecting the anomalies by an data annotator because it will *automate* the process using *Deep Learning techniques*.

Contents

Chapter 1

Introduction about the Industry

Oracle is an American multinational computer technology corporation headquartered in Austin, Texas. The company was formerly headquartered in Redwood Shores, California until December 2020 when it moved its headquarters to Texas. The company sells database software and technology, cloud engineered systems, and enterprise software products—particularly its own brands of database management systems. In 2020, Oracle was the second-largest software company in the world by revenue and market capitalization. The company also develops and builds tools for database development and systems of middle-tier software, Enterprise Resource Planning (ERP) software, Human Capital Management (HCM) software, Customer Relationship Management (CRM) software (AKA customer experience), Enterprise Performance Management (EPM) software, and Human Capital Management (HCM) software Customer Relationship Management (CRM) software (AKA customer experience) Enterprise Performance Management (EPM) software Supply Chain Management (SCM) software.

Oracle Cloud Infrastructure (OCI) is a deep and broad platform of public cloud services that enables customers to build and run a wide range of applications in a scalable, secure, highly available, and high-performance environment. The benefits of OCI include -

- *Autonomous services*
  OCI is the exclusive home of Oracle Autonomous Database and its self-repairing, self-optimizing autonomous features. Leveraging machine learning to automate routine tasks, Autonomous Database delivers higher performance, better security, and improved operational

efficiency, and frees up more time to focus on building enterprise applications.

- *Reduce costs and enhance performance*
  Oracle Cloud Infrastructure is built for enterprises seeking higher performance, lower costs, and easier cloud migration for their existing on-premises applications, and better price-performance for cloud native workloads. Read how customers have moved from AWS to Oracle Cloud Infrastructure, substantially reducing their costs and enhancing their compute platform performance.

- *Best support for hybrid architectures*

- *Easily migrate enterprise apps*
  Traditional, on-premises workloads that enterprises rely on to run their business are easier to migrate to Oracle Cloud. Designed to deliver bare-metal compute services, network traffic isolation, and the only service-level guarantees for performance, Oracle Cloud enables rapid migration and faster time to innovation. Build new value around migrated applications faster with Autonomous Database, data science tools, and our cloud native development tools. Deploy your cloud applications and databases anywhere with a wide choice of options, ranging from our public regions to edge devices. In addition to our public cloud region, we offer full private Dedicated Regions in customers data centers, edge-computing Oracle Roving Edge devices, and our blazingly fast Oracle Exadata Cloud@Customer, with Autonomous Database service delivered behind your firewall. With full support for VMware environments in the customer tenancy as well, Oracle offers cloud computing that works the way you expect.

Oracle Cloud Infrastructure Products:

- *Analytics*
  Oracle Analytics uses built-in machine learning and artificial intelligence to analyze data across your organization so you can make smarter, more informed decisions, faster.

- *Application development*
  Code less and build more web, mobile, and cloud native applications faster with Oracle Cloud's open and integrated application development platform.

- *Applied software technologies*
  Transformational technologies like AI, blockchain, machine learning, data science, and digital assistants are revolutionizing entire industries by integrating these services into enterprise applications or using Oracle's ready-to-use, modern SaaS applications that embed these applied software technologies into your fundamental business processes.

- *Compute*
  OCI built our infrastructure to deliver the performance, scalability, and services needed to run the most demanding enterprise workloads and the most modern cloud native applications. We offer a compute service designed for every workload, delivering new levels of cloud performance for HPC and enterprise applications with our bare metal Intel and nVidia GPU services, custom and dynamically elastic compute shapes on our AMD Epyc services, as well as cloud native services for Kubernetes and serverless functions.

- *Database*
  Oracle Autonomous Database is a unified database offering that brings Oracle database technology into an automated, self-healing platform. With support for multiple modes and data structures, we automate all routine database tasks, saving countless hours of manual, human effort and ensuring higher performance, reliability, security, and operational efficiency.

- *Integration*
  Accelerate digital transformation initiatives with Oracle integration. Get a jumpstart with preintegrated connectivity to Oracle and third-party SaaS and on-premises apps, run-ready process automation templates, and an intuitive visual app builder for rapid web and mobile app development.

- *Observability and management*
  Eliminate the human effort required to monitor and manage applications and infrastructure. Oracle Cloud Observability and Management services work with the entire operational data set for managing on-premises environments, Oracle Cloud environments, and hybrid and multicloud environments.

- *Networking, connectivity, and edge services*
  OCI networking services offer Layer 2 isolation of your tenancy to prevent 'noisy neighbors' from disrupting your workloads and offers highly

customizable virtual cloud networks (VCN) and connectivity services that extend your IT infrastructure without the massive network egress services common in public clouds.

- *Security, identity, and compliance*
  Concerns about security shouldn't keep you from the cloud, it should move you toward it. Implement and manage consistent security, identity, and access management policies across the hybrid data center while maintaining compliance.

- *Storage*
  Cloud storage doesn't have to be confusing, surprisingly expensive, or slow. Oracle Cloud delivers consistent, low-cost cloud storage options, including on-demand local, object, file, high IOPS block, and standard block storage services, archive storage, and secure data transfer services.

Chapter 2

Training Schedule

Internship duration: **June 1, 2022 - July 26, 2022**

1. *June 1, 2022*
   IT Setup - installation of all the required company software and Configuration of the Device.

2. *June 2, 2022 - June 3, 2022*
   Oracle Cloud Infrastructure Administration Essentials Training.

3. *July 6, 2022*
   Mid term demo and presentation of the project work done till then.

4. *July 22, 2022*
   Final project demo and presentation of the entire work completed.

5. *July 23, 2022*
   KT(Knowledge Transfer) Session with the team.

6. *July 24, 2022*
   Meeting with mentor and manager.

7. *July 25, 2022 - July 26, 2022*
   System retiring and uninstalling of all company software.

Weekly Description of work

Week 1

The first week started off with the on-boarding process and installation of various software and setting up Devices. We were introduced to what the company does and many senior folks of Oracle Cloud Infrastructure also talked to us. Followed by Oracle Cloud Infrastructure Essentials training. Was introduced to my team with which I would be working for the next few weeks.

Week 2

I was shared the code for the ART Tool . Went through the code and got all my doubts cleared. I also went through documentation of various technologies that were being used in the service.

Week 3

In this week I was exploring methods such as Time Seires data analysis and Anomaly Detection. Was devising ways in which I could implement the same the Anomaly Detection feature. Various libraries and frameworks were also explored in addition to implementing a workflow that would reduce the latency.

Week 4

After the theoretical aspects were explored and learnt, low level changes were being made. During this process, there was a complication with respect to Points selection through the graph that was plotted using the raw time series data. As a result, this problem was solved by using some framework. I prepared a detailed report stating this feature implementation. Suggestions were also made on new approaches.

Week 5

I Completed the Application Building part, All the required feature for the App was working perfectly such as selecting anomaly points manually and saving them in the databases. Then, I was assigned with the task of Building the Deep Learning Anomaly Detection model and to integrate it with

the application to automate the process.

Week 6

Worked on and coming up with an algorithm that would implement Anomaly Detection After testing various State of the Art Deep Learning models on Anomaly Detection.

Week 7

Worked on cleaning the code and fixing bugs. Had to Integrate this project into the ART tool where various AI Tasks were running on various datasets.

Week 8
In the final week, I was preparing the presentation for the final demonstration with the larger team. Made the application working perfectly and it was deployed on the web. Presented to my manager, mentors and other members of the team. Off-boarding took place after the final presentation.

Chapter 3

Introduction

3.1 Anomaly Detection in Time Series Data

Anomaly detection is the process of identifying the anomalies in a time series dataset. Time series dataset with anomalies leads to inconsistent results during forecasting. We will use the Isolation Forest algorithm to train a time series model. We will also plot a line chart to display the anomalies in our dataset.

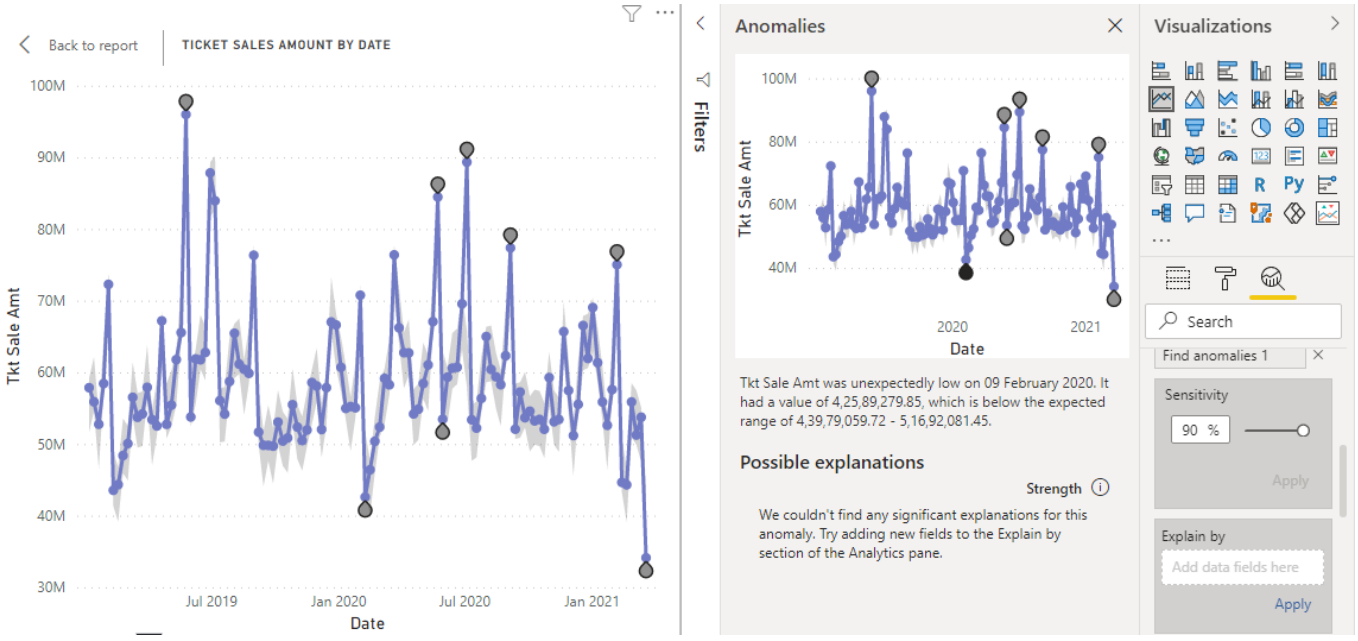

Figure 3.1: Anomaly Detection

3.2 Prerequisites for using the Anomaly Detection Tool

Overview

It is an Anomaly detection labeling tool, specifically for multiple time series (one time series per category). It is used for creating labeled data for anomaly detection models. Users can select points on a time series (created by the tool after uploading data) and inspect them by looking at the behavior of other time series at the same time range or the uploaded raw data.

User needs to have:

- Data set on which the Anomaly Detection should be performed.
- Setup of the tool installed on the System to run the tool locally. Cloud Version of the tool required *OCI Authentication Token*.
- Account in the proper compartment in OCI(Oracle Cloud Infrastructure) Console.

  (please refer to `https://docs.oracle.com/en-us/iaas/Content/API/Concepts/apisigningkey.htm#Required_Keys_and_OCIDs` for more details).

3.3 Working of the Anomaly Detection Tool

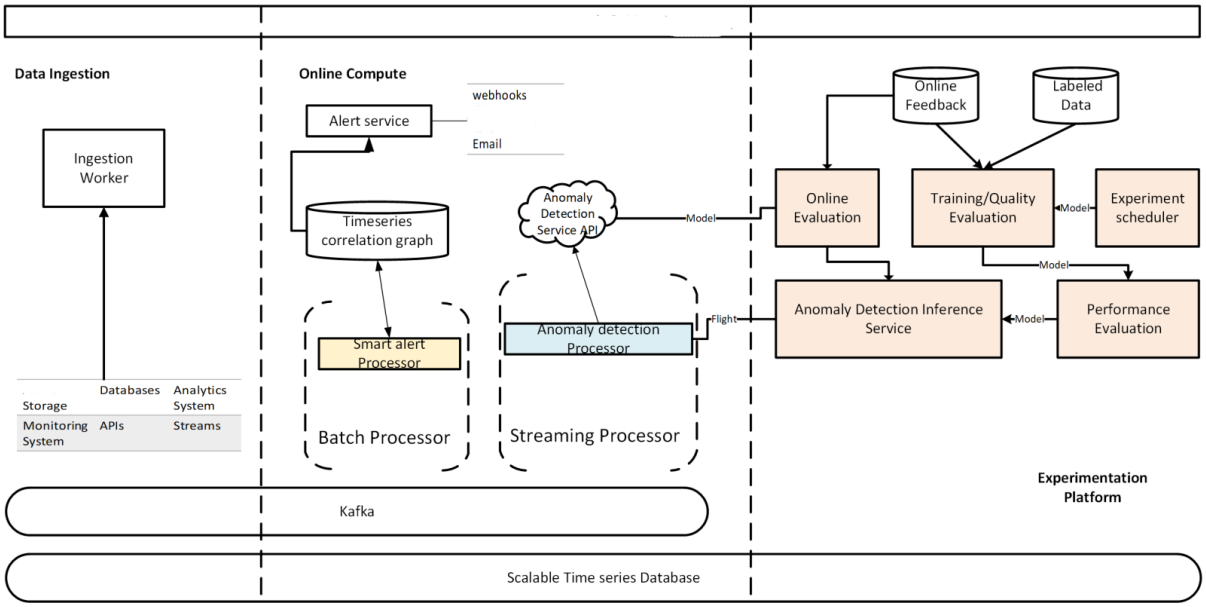

Figure 3.2: Flowchart of the AD Tool

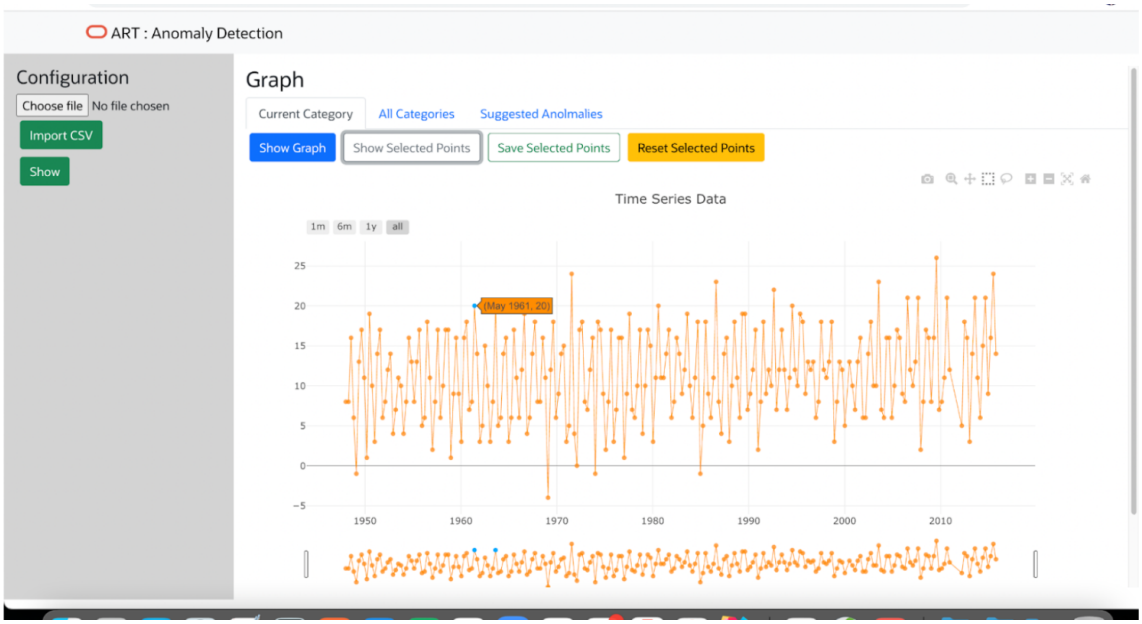

Figure 3.3: User Interface of the Application.

3.4 Instruction of use

- Import time series CSV file. Structure should be: date, category, value
- Import raw data time series CSV file. If the original time series is an aggreation over time windows, this time series is the raw values themselves. This way we could dive deeper into an anomalous value and see what it is comprised of.
  Assumed structure:
  - date (%Y-%m-%d %H:%M:%S) format
  - category
  - value
  - Select category (if exists)
- Select time range on slider
- Inspect your time series:
  - Click on one time range on the table below the plot to see raw data on this time range
  - Open the "All Categories" tab to see how other time series behave on the same time range.
- Select points on plot that look anomalous.
- Click "Add selected points" to add the marked points to the candidate list.
- Once you decide that these are actual anomalies, save the resulting table to csv by clicking on "Download labels set" and continue to the next category.

3.5 Notable features

- *The labeling window*
  Here users can:
  - Load raw/time-framed data
  - Select category (from the data)
  - Interpolate the missing points in the data.
  - Select Graph break points.
  - Time frame to analyze the data
- *Time series labeling*
  Drag and select the points from the cursor to label the data
- *Selected points table view*
  Here users can see the desired point in table format.
- View raw data for the selected window (if it exists).
- Compare this category with others over time.
- Observe the changes in distribution between categories
- This is useful to understand whether an anomaly was univariate or multivariate.
- Find proposed anomalies using the Twitter AnomalyDetection package.

Chapter 4

Work Done

4.1 Initial setup

Following initial requirements were needed to be done at first from my side so as to setup for the Anomaly Detection tool:

- Git setup `https://confluence.oci.oraclecorp.com/pages/viewpage.action?spaceKey=HRZN&title=GIT+Setup+for+Windows`
- Repository for ART `https://bitbucket.oci.oraclecorp.com/projects/OCAIS/repos/ocasbt-art/browser`
- All MLTask in ART `https://confluence.oci.oraclecorp.com/pages/viewpage.action?pageId=980229129`
- Basic Tool to setup:
  - MongoDB and Robo 3T (now Studio 3T Free)
  - Docker
  - Git Terminal
  - VS Code and IntelliJ Idea
- Understand the code and ART tool funtionalities. Run all the commands after setup. Run on different dataset models to the on art.
- Setup of Docker, VSCode, Git, Robo 3T (Studio 3T) and cloned the ocasbt-art to my machine.
- Explore the art tool and uploaded the data set, imported into ART-tool, then downloaded (export) them after reviewing some images using CLI.

4.2 Algorithm/Procedure/Methodology

Firstly, Data set should be uploaded on the application and then the plot will be render on the UI using the *plotly.js*.

The Application Interface is build with the *React framework* and the back end with *Spring boot*.

After the rendering of the data annotator can annotate the anomalies manually or using the in-built *Anomaly Detection* feature that was made during the project.

After annotation of the anomaly the points can be saved to the databases (I used *MongoDB* database) and can be reviewed later.

Techniques used for Anomaly Detection in Time Series data.

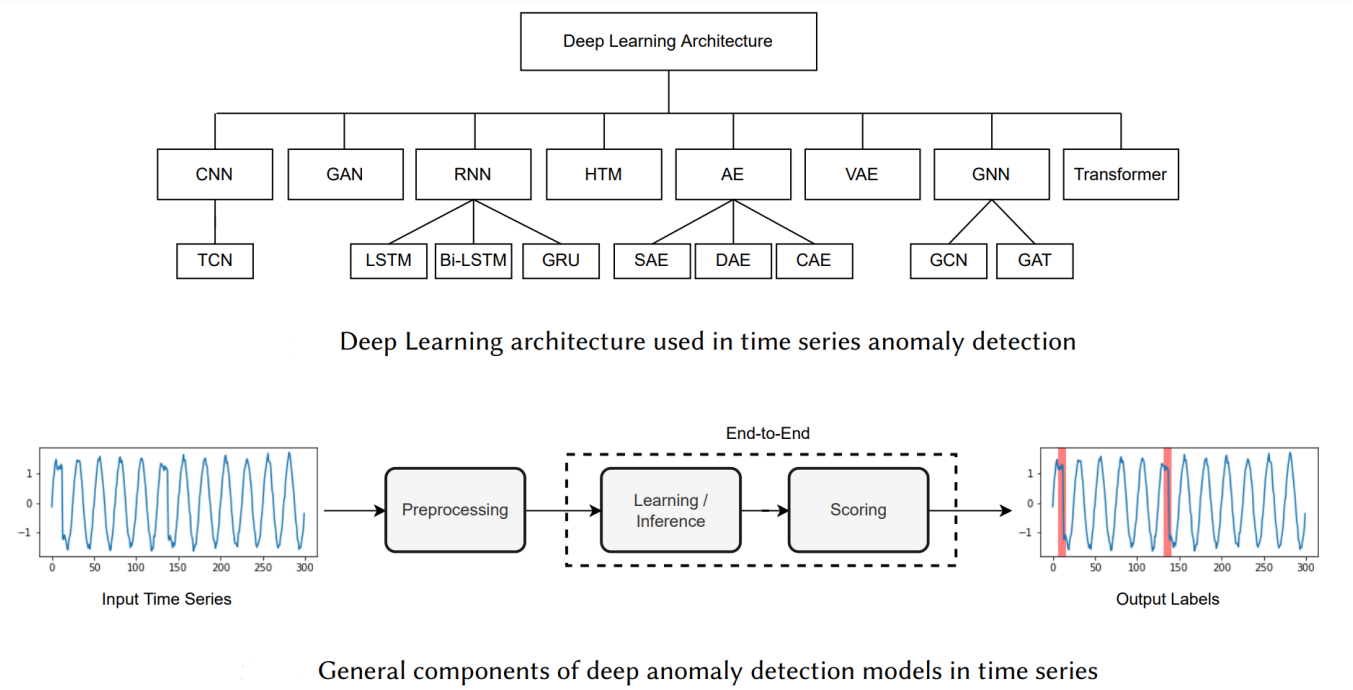

Figure 4.1: Deep Learning Models for Anomaly Detection on Time series

In RNN(Recurrent Neural Network) used LSTM model for Anomaly Detection

Figure 4.2: Algorithm for Anomaly Detection

Interpolate missing points

There are different interpolation techniques that could be used to impute the data missing in the time series. However, interpolation looks more like a data preparation or correction task and may not be functionally aligned with ART tool.

Interpolation methods for time series data

- *Linear interpolation*
  Linear interpolation is the most straightforward and commonly used interpolation method. It comes naturally when we have two points, we connect them with a straight line to fill out the missing information in between.
  When we have fewer points, the pursuit for higher accuracy prevails. And the errors between the interpolated values and their true values have a larger impact on the analysis. Thus, more interpolation methods are created in order to meet such demands. One of the examples is polynomial interpolation.

- *Polynomial interpolation*
  Polynomial interpolation assumes that the points are samples taken from a polynomial curve. We can describe three or more points by a linear function if all these points lie on the same line. Hence, N points can be described by a polynomial of no more than N-1 degrees. Consequently, there must exist a polynomial with the least number of degrees that describes all the points. This polynomial, given the observed points, is called a Lagrange polynomial.

- *Piecewise cubic hermite interpolation*
  Here we say, polynomials that pass through all points and at least have a continuous first derivative function (thus no jump in slopes and no corners at every point) are called Hermite interpolation polynomials.

- *Pchip, cubic spline, makima*
  Piecewise Cubic Hermite Interpolation Polynomials are abbreviated as "pchip". Many softwares exclusively refer to interpolation polynomials that only have a continuous first derivative function as pchip. And if we require an additional continuous second derivative function, then it becomes a cubic spline. The difference between these two methods is how the derivatives at the points are defined. And there are other ways to definite it too and one of them is makima.

Input the missing values

- *Mean imputation*
  This technique imputes the missing values with the average value of all the data already given in the time series.

- *Median imputation*
  In this technique, we replace the missing values in the data with the median value of the data.

- *Last Observation Carried Forward (LOCF)*
  According to this technique, the missing value is imputed using the values before it in the time series.

- *Next Observation Carried Backward (NOCB)*
  According to this technique, the missing values are imputed using an immediate value ahead of them.

Anomaly Detection Approaches

- *Model-based*
  In this kind of approach a model of the data is built. The objects that do not fit very well are considered as anomalies. For example let us assume that a model is built of certain data which is in the form of cluster. Then anomalies are those data objects that do not strongly belong to the cluster built.

- *Distance-based*
  This approach is based on the proximity's. Consider a 2D or 3D scatter plot all the data objects are in one proximity. But Anomalous objects are away from them.

- *Density-based*
  Density of objects is easy to compute especially if a proximity measure between objects is available. Low density objects are those that relatively distant from neighbours.

There are three basic categories for anomaly detection:

- **Supervised Anomaly Detection**
  In this supervised learning there must a training set for both data objects and expected anomalous objects. We have to observe that there can be more than one anomalous class.

- **Unsupervised Anomaly Detection**
  For situations where class labels are not available. We can give a score for each object that shows the degree to which the instance is anomalous. If there are many anomalies present which are similar to each other, then we can group them as normal group or the outliers score is low.

- **Semi Supervised Anomaly Detection**
  Sometimes when there is training data with labelled normal objects and score given, but has no anomalous objects, then we can implement the semi supervised anomaly detection to find the anomalies.

Anomaly detection techniques in time series data

1. *Classification and Regression Trees (CART)*
   We can utilize the power and robustness of Decision Trees to identify outliers/anomalies in time series data. Isolation Forest detects anomalies purely based on the fact that anomalies are data points that are few and different. The anomalies isolation is implemented without employing any distance or density measure.

2. *Detection using Forecasting*
   Anomaly detection using Forecasting is based on an approach that several points from the past generate a forecast of the next point with the addition of some random variable.

3. *Clustering-based anomaly detection*
   The approach is pretty straightforward. Data instances that fall outside of defined clusters could potentially be marked as anomalies. We can use k-means clustering.

4. *Autoencoders*
   Autoencoders are an unsupervised technique that recreates the input data while extracting its features through different dimensions. Autoencoder techniques can perform non-linear transformations with their non-linear activation function and multiple layers. It's more efficient to train several layers with an autoencoder, rather than training one huge transformation with PCA. The autoencoder techniques thus show their merits when the data problems are complex and non-linear in nature.

4.3 Technologies/Tools used

4.3.1 Software Languages and Packages

HTML

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. HTML was used in the project for rendering all the frontend pages of the *Anomaly Detection (AD) Tool*. Its used along with React

CSS

Cascading Style Sheets or CSS is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS was used in the project so as to enhance the User Interface and the User Experience of the plugin. It used with the compiled version as scss with React.

JavaScript

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript was used in the project for maintaining the entire functionality of the AD Tool and the flow of data from frontend to backend and database.

Java

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere, meaning that compiled Java code can run on all platforms that support Java without the need to recompile. It used to write the Backend of the *AD Tool* with *Spring Boot*

4.3.2 Frameworks and Libraries

- *React JS*
  A JavaScript library for building user interfaces.React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

- *Plotly JS*
  Use *react-plotly.js* to embed D3 charts in your React-powered web application. This React component takes the chart type, data, and styling as Plotly JSON in its data and layout props, then draws the chart using *Plotly.js*.

- *Spring Boot*
  Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run". We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss.

Download

- GitHub - `https://github.com/oracle/oci-typescript-sdk`

- ReactJS - `https://reactjs.org/`

- Spring Boot - `https://spring.io/projects/spring-boot`

Requirements

To use the React and PlotlyJS you must have the following:
React 18 supports all modern browsers (Edge, Firefox, Chrome, Safari, etc). If you support older browsers and devices such as Internet Explorer which do not provide modern browser features natively or have non-compliant implementations, consider including a global polyfill in your bundled application.

Versions Supported

React Support version is v18.x. Tensorflow version v2.10.x. and Spring Boot version v2.0.x

4.3.3 Tools and Softwares

OCI Console

OCI Console is a browser based service provided by Oracle Cloud Infrastructure. I used the OCI Console during my entire project so as to have a view from OCI side as well. Please refer the following links for more documentation on how to signup, login and use OCI Console.

- `https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/signingin.htm`
- `https://docs.oracle.com/en-us/iaas/Content/GSG/Concepts/console.htm`

Code Editors

- *IntelliJ IDEA*
  Download - `https://www.jetbrains.com/idea/`
  IntelliJ IDEA is an integrated development environment written in Java for developing computer software written in Java, Kotlin, Groovy, and other JVM-based languages. It is developed by JetBrains and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition.
- *Visual Studio Code*
  Download - `https://code.visualstudio.com/`
  Visual Studio Code is a lightweight source code editor which comes in with a whole set of IDE features like built-in git, intellisense, debugger and a huge range of installable extensions. The source code of the project was written using this editor.

BitBucket

Bitbucket is a Git-based source code repository hosting service owned by Atlassian. Bitbucket was used as the Version Control System for maintaining my entire project.

Jira

Jira is a proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management. Its used as maintaining the requirements and to trace the feature of the application.

Chapter 5

Results and Analysis

5.1 Results

- The project was successfully completed with all the code, documentation, installation steps, usage demo, possible future work, etc. being provided within the given duration of the internship.

- The end product of the internship had most of the requirements completely done which could be used at development stages by internal users, but still having work to be done before deploying to production for use by everyone.

- The final intern demo was successfully conducted where I explained the whole project to the Oracle AI Services team and had also shown a live demo of the work, briefing about the features incorporated, the methodology adopted, complete flow of the project and also some the possible future steps.

5.2 Analysis

- The project required understanding of a lot of technologies like Time Series Data Analysis, Web technologies, ReactJS, Docker, Rendering large Data, Categorical Plot Generation from raw data, Product Deployment, unit-testing and also Oracle technologies like OCI-Infrastructure etc.

- Understanding each and every technology's functionality and integrating those together to build the final working product was definitely a challenging work which required proper focus and complete clarity.

- A lot of work needs to be done before coding in reality - for any problem/feature/bug fix, it is required to analyze different possible approaches and discover in detail the affects, use-cases etc. so as to find the best feasible solution without violating the basic principles and the best practices as well.
- The project made me learn how to understand and value the qualitative requirements and convert those into quantitative outcomes.
- The internship provided me an opportunity to network and interact with the senior employees of the company, learn from their experiences and take in their valuable thought process which helped me a lot during my entire internship and improved my skills to a lot of extent.

Chapter 6

Conclusion

The internship helped me get a lot of exposure to how software development is being done in the real world including so many people specialized in different domains and how every member brings in his/her expertise to contribute to the completion of any product. The internship not only helped me improve my technical skills but also made me realize the importance of networking with other people, expressing out our own thoughts to others in the most feasible way and also the important presentation skills required for an individual. The intern project helped me understand the importance of many factors in any production level product which otherwise are neglected while done at individual level or as side-projects.

This Project contributed by me impacted in positive manner for the organization.

Overall, the internship was a great experience for me, providing many valuable life lessons which will definitely play a very important role in my whole life and for my overall growth.

References

[1] Oracle Corporation Wikipedia - https://en.wikipedia.org/wiki/Oracle_Corporation

[2] Oracle Cloud Infrastructure - https://www.oracle.com/in/cloud/

[3] Oracle Cloud Infrastructure Documentation - https://docs.oracle.com/en-us/iaas/Content/API/Concepts/sdkconfig.htm

[4] HTML Wikipedia - https://en.wikipedia.org/wiki/HTML

[5] CSS Wikipedia - https://en.wikipedia.org/wiki/CSS

[6] JavaScript MDN - https://developer.mozilla.org/en-US/docs/Web/JavaScript

[7] Docker - https://www.docker.com/

[8] Tensorflow - https://www.tensorflow.org/

[9] Plotly - https://plotly.com/javascript/

[10] React - https://reactjs.org/

[11] Java - https://www.java.com/en/

[12] Spring Boot - https://spring.io/projects/spring-boot

[13] MongoDB -https://www.mongodb.com