**Industrial Internship Report on**

**”Forecasting of Smart City Traffic System”**

**Prepared by**

**K SAI HARSHITHA**

|  |
| --- |
| *Executive Summary* |
| This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).  This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.  My project was about Forecasting of Smart City Traffic System where the dataset is provided under various conditions of Traffic at distinct Junction points. We will be counting the number of vehicles at a unique timestamp (date, hour, seconds, minute, quarter time), arranging them in descending order so that we will be having the highest vehicle count at that time stamp lowest at the end.  This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship. |

**TABLE OF CONTENTS**

[1 Preface 3](#_Toc139702806)

[2 Introduction 4](#_Toc139702807)

[2.1 About UniConverge Technologies Pvt Ltd 4](#_Toc139702808)

[2.2 About upskill Campus 8](#_Toc139702809)

[2.3 Objective 9](#_Toc139702810)

[2.4 Reference 9](#_Toc139702811)

[2.5 Glossary 10](#_Toc139702812)

[3 Problem Statement 11](#_Toc139702813)

[4 Existing and Proposed solution 12](#_Toc139702814)

[5 Proposed Design/ Model 13](#_Toc139702815)

[5.1 High Level Diagram (if applicable) 13](#_Toc139702816)

[5.2 Low Level Diagram (if applicable) 13](#_Toc139702817)

[5.3 Interfaces (if applicable) 13](#_Toc139702818)

[6 Performance Test 14](#_Toc139702819)

[6.1 Test Plan/ Test Cases 14](#_Toc139702820)

[6.2 Test Procedure 14](#_Toc139702821)

[6.3 Performance Outcome 14](#_Toc139702822)

[7 My learnings 15](#_Toc139702823)

[8 Future work scope 16](#_Toc139702824)

# Preface

Summary of the whole 6 weeks’ work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

# Introduction

## About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various**Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end**etc.



1. UCT IoT Platform **(****)**

**UCT Insight** is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

* It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
* It supports both cloud and on-premises deployments.

It has features to  
• Build Your own dashboard  
• Analytics and Reporting  
• Alert and Notification  
• Integration with third party application(Power BI, SAP, ERP)  
• Rule Engine

1. **Smart Factory Platform (****)**

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

* with a scalable solution for their Production and asset monitoring
* OEE and predictive maintenance solution scaling up to digital twin for your assets.
* to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and improve them.
* A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost, and money.

1.  based Solution

UCT is one of the early adopters of LoRAWAN technology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

1. Predictive Maintenance

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful lifetime of various Machines used in production process.



## About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

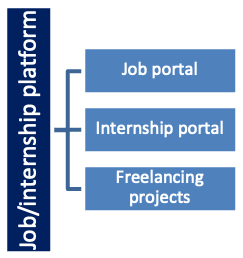
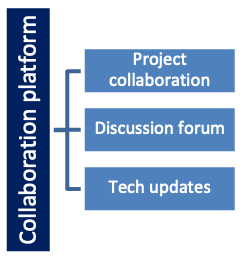
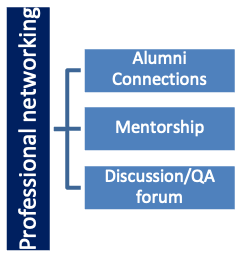
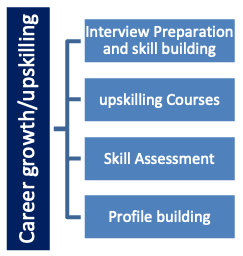
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable, and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

<https://www.upskillcampus.com/>

upSkill Campus aiming to upskill 1 million learners in next 5 year



## The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

## Objectives of this Internship program

The objective for this internship program was to

 ☛ get practical experience of working in the industry.

 ☛ to solve real world problems.

 ☛ to have improved job prospects.

 ☛ to have Improved understanding of our field and its applications.

 ☛ to have Personal growth like better communication and problem solving.

## Reference

* Python Data Science Book by “John Mueller”, 2015 Edition
* Got Information from “Springer” Link
* Guidance by Traffic Detection System

## Glossary

|  |  |
| --- | --- |
| Terms | Acronym |
| Numpy | An Open First, I can’t library, that’s used in almost every field of Computer science and engineering |
| Pandas | A python Library used for working with data set |
| Decision tree | A type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered |
| Sklearn | An open-source data analysis, library function and the gold standard for machine learning in python ecosystem. |
| Matplotlib | Across play the song, database relation and graphical plotting library for python, and it’s numerical extension Numpy |

# Problem Statement

The government wants to transform a city into a smart city. The vision is to convert it into a digital and intelligent city to improve the efficiency of services for the citizens. One of the problems faced by the government is traffic.

The government wants to implement a robust traffic system for the city by being prepared for traffic peaks. They want to understand the traffic patterns of the four junctions of the city. Traffic patterns on holidays, as well as on various other occasions during the year, differ from normal working days. This is important to consider for your forecasting.

Traffic places a major role in every citizen life, people who are travelling from home to office and through the busiest areas. The strain generated by traffic jams, sleep, and workouts induced by time spent in traffic. Since the motorist cannot see the entire traffic system directly, hence, this traffic system must be anticipated to make the motorist to choose the right path and the subsequent impact on the environment, as well as to implement smart transport system. To overcome this, we are using Concepts like Decision Tree, Pandas, NumPy, Matplotlib, sklearn to predict traffic pattern in a smart city. Analysis results show that increase in the number of junctions of the city can alleviate problem being faced on the road by commuters.

# Existing and Proposed solution

In big-data-driven traffic flow prediction systems, the robustness of prediction performance depends on accuracy and timeliness. This paper presents a new MapReduce-based nearest neighbor (NN) approach for traffic flow prediction using correlation analysis (TFPC) on a Hadoop platform. We develop a real-time prediction system including two key modules, i.e., offline distributed training (ODT) and online parallel prediction (OPP). Moreover, we build a parallel K - nearest neighbor optimization classifier, which incorporates correlation information among traffic flows into the classification process. Finally, we propose a novel prediction calculation method, combining the current data observed in OPP and the classification results obtained from large-scale historical data in ODT, to generate traffic flow prediction in real time. The empirical study on real-world traffic flow big data using the leave-one-out cross validation method shows that TFPC significantly outperforms four state-of-the-art prediction approaches, i.e., autoregressive integrated moving average, Naïve Bayes, multilayer perceptron neural networks, and NN regression, in terms of accuracy, which can be improved 90.07% in the best case, with an average mean absolute percent error of 5.53%. In addition, it displays excellent speedup, scaleup, and size up.

By looking above summary, they used TFPC, ODT, NN concepts which are complex to learn and understand,

Who is simplify this, we are using the very simple concepts like Decision Tree, Pandas, NumPy, Matplotlib, Sklearn. Accordingly, we will be developing a module such that according to the timestamp we will classifies, the number of vehicles passing through the junction point.

## GitHub Link:https: https://github.com/Saiharsithareddy/Upskill-internship/tree/main

## Report submission (GitHub link): https://github.com/Saiharsithareddy/Upskill-internship/tree/main

## High Level Diagram (if applicable)

A graph of red rectangular bars

Description automatically generated

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

## Low Level Diagram (if applicable)

A graph of a graph

Description automatically generated

# Performance Test

This is very important part and defines why this work is meant of Real industries, instead of being just academic project.

Here we need to first find the constraints.

How were those constraints taken care in your design?

What were test results around those constraints?

Constraints can be e.g., memory, MIPS (speed, operations per second), accuracy, durability, power consumption etc.

In case you could not test them, but still, you should mention how identified constraints can impact your design, and what are recommendations to handle them.

## Test Plan/ Test Cases

train\_module['DateTime'] = pd.to\_datetime(train\_module['DateTime'])

test\_module['DateTime'] = pd.to\_datetime(test\_module['DateTime'])

test\_module.info()

In [ ]:

test\_module['Weekday'] = [datetime.weekday(date) for date in test\_module.DateTime]

test\_module['Year'] = [date.year for date in test\_module.DateTime]

test\_module['Month'] = [date.month for date in test\_module.DateTime]

test\_module['Day'] = [date.day for date in test\_module.DateTime]

test\_module['Time'] = [((date.hour\*60+(date.minute))\*60)+date.second for date in test\_module.DateTime]

test\_module['Week'] = [date.week for date in test\_module.DateTime]

test\_module['Quarter'] = [date.quarter for date in test\_module.DateTime]

*# Creating features from DateTime for test data*

train\_module['Weekday'] = [datetime.weekday(date) for date in train\_module.DateTime]

train\_module['Year'] = [date.year for date in train\_module.DateTime]

train\_module['Month'] = [date.month for date in train\_module.DateTime]

train\_module['Day'] = [date.day for date in train\_module.DateTime]

train\_module['Time'] = [((date.hour\*60+(date.minute))\*60)+date.second for date in train\_module.DateTime]

train\_module['Week'] = [date.week for date in train\_module.DateTime]

train\_module['Quarter'] = [date.quarter for date in train\_module.DateTime]

## Test Procedure

* Here, we are planning to work with the predefined libraries found in the python such as the NumPy, pandas and matplotlib using the google Colab.
* Simply import the required packages to work with them.
* Create a user defined function that will calculate the unique time from the given data set.
* Here we are using DecisionTreeClassifier which is available in sklearn modules, so import them.
* Read the given Data set initially and create the train and test modules to compare the outcomes which we got later.
* We’ll be having Test and Train Case
* Such that everything will be in descending order (Timestamps)
* Accordingly, time stamp scenario will be arranged in a way highest vehicle count will be in first row and the lowest will be at the end.

## 6.3 Performance Outcome



# My learnings

UpSkill campus is the first organization that provided me a free internship. I took the course data science and machine learning where I learned many library functions, implementation of data and data handling, how to access the CSV data files into my machine learning model. I done a project on Forecasting of smart city traffic systems provided by Upskill, this helped me a lot to know about the data, cleaning the data, implementation of various concepts like this is NumPy, pandas, Matplotlib, decision tree, Sklearn. By these concepts and with this project I developed my self-confidence, and I can deal much more projects. The credit is completely for UpSkill Campus. Thanks for providing this internship. I hope in future we continue.

# Future work scope

The world is rapidly becoming more interconnected and technology-driven, and cities are no exception. Smart cities are emerging as a result, utilizing the power of artificial intelligence (AI) and other advanced technologies to improve the quality of life for their citizens. Smart traffic management systems are a key component of this new urban landscape, allowing cities to manage traffic flow and reduce congestion.

Smart traffic management systems use AI to analyze data from sensors and cameras to detect patterns in traffic flow. This data is then used to optimize traffic signals, route traffic away from congested areas, and even adjust speed limits to improve safety. The result is a more efficient and safe flow of traffic, reducing the time and fuel wasted in traffic jams.

Smart traffic management systems also have the potential to reduce air pollution and improve public health. By optimizing traffic flow, these systems can reduce the amount of time cars spend idling in traffic, which in turn reduces emissions. Additionally, the data collected by these systems can be used to identify areas of high pollution and inform city planners of the need for improved public transportation or other measures to reduce emissions.

Finally, smart traffic management systems can also improve public safety. By providing real-time information on traffic conditions, these systems can alert drivers to dangerous situations, such as accidents or bad weather. This can help reduce the number of accidents and improve the safety of drivers and pedestrians alike.

Smart city AI and smart traffic management systems are quickly becoming an essential part of urban life. By improving traffic flow, reducing air pollution, and improving public safety, these systems are helping to make cities more livable and efficient. As the technology continues to develop, cities will be able to reap even more benefits from these systems.