"Data Driven Framework for Real-Time Machine Learning Based Prediction on Vehicle On-Board Diagnostic(OBD) Data."

A

Project Report

submitted in partial fulfillment of the

requirementsfor the award of the degree of

MASTER OF TECHNOLOGY

in

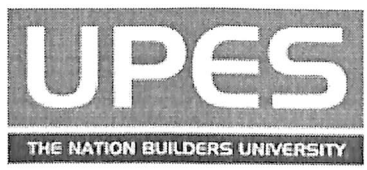
ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWROK

by

Name	Roll No.
Yashkumar U Vasa	R102214011

under the guidance of

Prof. Vishal Kaushik



Department of Computer Science & Engineering

Centre for Information Technology

University of Petroleum & Energy Studies

Bidholi, Via Prem Nagar, Dehradun, UK

April – 2016

The innovation driven
E-School

CANDIDATE'S DECLARATION

I hereby certify that the project work entitled "***Data Driven Framework for Real-Time Machine Learning Based prediction on vehicle On-Board Diagnostic(OBD) data***" in partial fulfillment of requirement for the award the degree of **MASTER OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE & ARTIFICIAL NEURAL NETWORK** and submitted to Department of Computer Science & Engineering at center for Information Technology, University of Petroleum & Energy Studies, Dehradun, in authentic record of my work carries out during a period from December, 2015 to April, 2016 under guidance the supervision of Prof. Vishal Kaushik

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

Yashkumar U Vasa
R102214011

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 08 / 04 / 2016

Prof. Vishal Kaushik
Project Guide

Dr. Amit Agrawal
Program Head – M.Tech Artificial intelligence & Artificial Neural Network.
Center for Information Technology
University of Petroleum & Energy Studies
Dehradun – 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

I wish to express our deep gratitude to our guide **Prof. Vishal Kaushik**, for all advice, encouragement and constant support he has given us through out our project work. This work would not have been possible without his support and valuable suggestions.

I am heartily thankful to my thesis advisor **MR. NRK Rao** and **Mr. Samit Prabhu** at **L&T Infotech**. The door to **Mr. NRK Rao** office was always open whenever I ran into a trouble spot or had a question about my research or writing.

I am heartily thankful to my course coordinator, Prof. Vishal Kaushik, for the precise evaluation of the milestone activities during the project timeline and the qualitative and timely feedback towards the improvement of the project.

I sincerely thank to our respected Program Head of the Department, Dr. Maneesh Prateek, for his great support in doing our project in **Area** at **CIT**.

I am also grateful Dr. Kamal Bansal and **Dr. Maneesh Prateek** Dean CoES, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Yashkumar U Vasa

R102241011

Abstract

Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is often feasible and cost-effective where manual programming is not. As more data becomes available, more ambitious problems can be tackled. As a result, machine learning is widely used in computer science and other fields. However, developing successful machine learning applications requires a substantial amount of "black art" that is hard to find in textbooks.

To tackle the problem of safety and vehicle fault detection, we have studied anomaly detection and fault diagnosis method for Automobile system based on Machine Learning (ML), Big Data, and Data Mining (DM) technology. In these method, the knowledge or model which is necessary for monitoring a Vehicle system is Semi-Automatically acquires from OBD system.

The Vehicle On-Board Diagnostics(OBD) System Telemetry Analytics Solution aims how the car dealerships, Automobile Manufactures, and insurance companies can leverage the capabilities of Machine Learning based Analytics to gain real-time and predictive insight on vehicle health and driving habits, to drive improvements in the area of customer experience, Research & Development. Real cases of Vehicle anomalies are considered, allowing assessing the effectiveness of algorithm (Principal Component Analysis), which showed to be effective in the case study where several telemetry channels tended to deliver outlier values.

Text Index: Machine Learning, Data-Mining, Anomaly Detection, OBD, Real-Time, Analytics, Vehicle, Health, Driving Method, Insurance.

Table of Contents

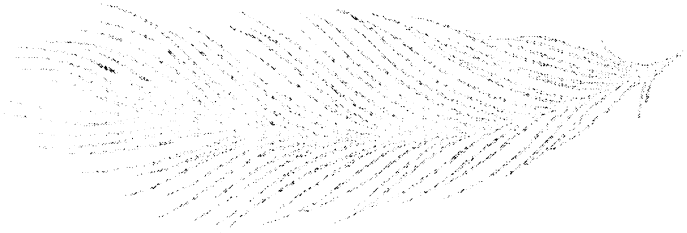
List of Figure

List of Table

"We Are Drowning in Information and Starving for Knowledge"

- John Naisbitt

CHAPTER 1. INTRODUCTION



1. Introduction

Insurance is a business built on data. Insurers analyze data to understand, evaluate and assume profitable risk. While data is the most valuable asset for insurers; actuaries, underwriters and other key stakeholders are hard to pressed to obtain the right data at the right time. Many are evaluating risk and making strategic decision based on only a sample set of historical and industrial data. These limited data source provide a myopic view and place tremendous burden on insurers.

1. Accurately assess risk across lines of business,
2. Reduce claim and underwriting leakage,
3. Detect and analyze fraud patterns,
4. Detect and report on regulatory non-compliance breaches,
5. Develop program such as "Pay-As-You-Drive" (PYAD) that requires easy access to telemetry data.

The aim of the project is to enable better management of vehicle health with help of Machine Learning:

1. Improves safety through use of diagnostics and prognostic to fix faults before they are issue.
2. Improves Availability through better maintenance scheduling.
3. Improves Reliability through through understanding of the current health of the system.
4. Leverages Automobile Manufacturer and insurance companies to gain real time vehicle health and driving habits.

Software for machine learning is widely available, and organizations seeking to develop a capability in this area have many options. The following requirements should be considered when evaluating machine learning:

1. Speed
2. Time to value
3. Model accuracy
4. Easy integration
5. Flexible deployment
6. Usability
7. Visualization

Let's review each of these in turn.

- **Speed**. Time is money, and quick package makes your extremely paid data scientists additional productive. sensible data science is usually reiterative and experimental; a project could need many tests, therefore tiny variations in speed translate to dramatic enhancements in potency. Given today's knowledge volumes, superior machine learning package should run on a distributed platform, therefore you'll be able to unfold the employment over several servers.
- **Time to value**. Runtime performance is just one a part of total time to worth. The key metric for your business is that the quantity of your time required to finish a project from knowledge bodily function to readying. In sensible terms, this implies that your machine learning package ought to integrate with standard Hadoop and cloud formats, and it ought to export prognostic models as code that you just will deploy anyplace in your organization.
- **Model Accuracy**. Accuracy matters, particularly once the stakes area unit high. For applications like fraud detection, tiny enhancements in accuracy will turn out

countless greenbacks in annual savings. Your machine learning package ought to empower your knowledge scientists to use all of your knowledge, instead of forcing them to figure with samples.

- **Easy integration**. Your machine learning package should co-exist with a posh stack of massive knowledge package in production. Ideally rummage around for machine learning package that runs on goods hardware and doesn't need specialized HPC machines or exotic hardware like GPU chips.
- **Flexible readying**. Your machine learning package ought to support a spread of readying choices, as well as co-location in Hadoop or in an exceedingly detached cluster. If cloud is a component of your design, rummage around for package that runs in an exceedingly kind of cloud platforms, like Amazon net Services, Microsoft Azure, and Google Cloud Platform.
- **Usability.** data scientists use many alternative package tools to perform their work, as well as analytic languages like R, Python, and Scala. Your machine learning platform ought to integrate simply with the tools your data scientists already use. additionally, well-designed machine learning algorithms embody time-saving options like the following:

1. Ability to treat missing data
2. Ability to remodel categorical data
3. Regularization techniques to manage complexity
4. Grid search capability for machine-driven test and learn
5. Automatic cross-validation (to avoid overlearning)

Visualization. triple-crown prognostic modeling needs collaboration between the data scientist and business users. Your machine learning software should provide business users with tools to visually judge the standard and characteristics of the prognostic model.

1.1 On-Board Diagnostics

On-board diagnostic systems (OBD) were developed within the 1980's to assist technicians diagnose and service the computerized engine systems of recent vehicles. a replacement generation of those systems known as OBD II is gift on 1996 and newer vehicles. These new systems, despite the type of auto, currently monitor normal components, use similar computer "language," and have similar criteria for evaluating the systems and indicating issues to the driving force and the repair technician.

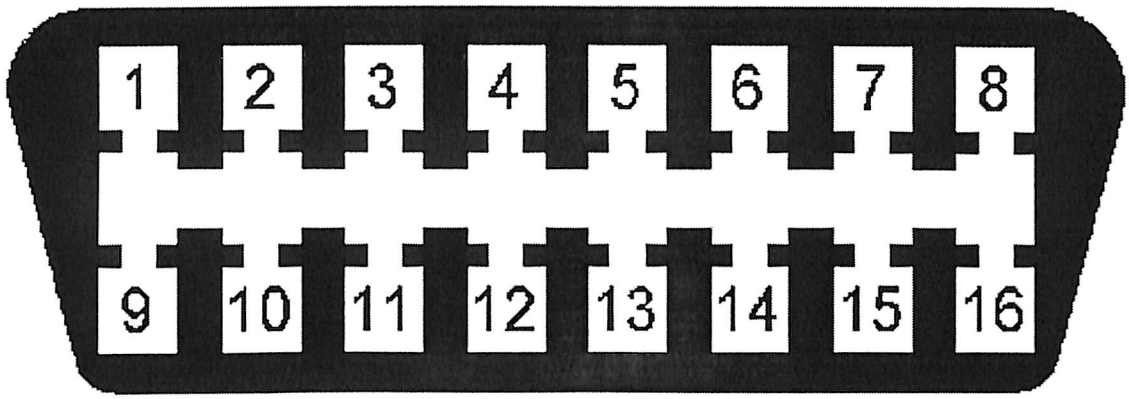


Figure 1: OBD - II Sensor Connector

Pin No.	Protocol/Use
Pin 2	J1850 Bus+
Pin 4	Chassis Ground
Pin 5	Signal Ground
Pin 6	CAN High(J-2284)
Pin 7	ISO 9141-2 K Lines
Pin 10	J1850 Bus
Pin 14	CAN Low (J-2284)
Pin 15	ISO 9141-2 Layer
Pin 16	Battery Power

Table 1: Pin Layout OBD II Sensor

OBD monitors the components that make up the emission system and key engine components. It can usually detect a malfunction or deterioration of these components before the driver becomes aware of the problem.

1.1.1 Benefits of On-Board Diagnostics System:

Cars and trucks are accountable for approximately half the pollution that causes smogginess and air pollution and contribute to global climate change. whereas it's true that modern cars emit less pollution than older vehicles, {they square measure they're} solely cleaner if their emission management systems are in operation properly.

The OBD II system will usually discover a vehicle malfunction before the motive force becomes conscious of the matter. Early detection and repair of malfunctions can result in fewer emissions and also the early repair of minor issues could stop a lot of vital and costlier engine issues that could develop if left unrepaired.

For example, a poorly performing spark plug can cause the engine to misfire, a condition generally unheeded by the driver, however one which will be detected by the OBD II system. This engine misfire can, in turn, quickly degrade the performance of the convertor and for good injury the catalyst. By responding to the check engine light (turned on by the OBD II system) in a timely manner, the motive force would be faced with a comparatively cheap

spark plug repair. However, while not OBD II detection, the motive force may well be two-faced with an upscale convertor repair additionally to the plug repair.

By storing the malfunction info within the computer's memory at the time it occurs, OBD II permits the service technician to a lot of accurately establish the matter and create the right repairs. this protects time for the repair technician, money for the consumer, and reduces air pollution.

1.1.2 OBD system Works

The OBD II system monitors a variety of engine condition and outputs while the car is being driven, the OBD II system detects a Problem with the emission control system, a dashboard light is illuminated indicating "*Check System*", A corresponding diagnostic trouble code is stored in the computer's memory documenting which emission control component us experiencing the problem, and under what condition. The repair technician will retrieve the diagnostic trouble code information from the computer using a computer scan tool. By using this information, a properly trained technician can more accurately find and fix the problem.

If the malfunction indicator light illuminates with a steady, continuous light, the vehicle operator should contact a repair technician and schedule a service visit. This is not an emergency situation, but the vehicle should be serviced soon. However, if the malfunction indicator light blinks or flashes, this indicates certain severe engine malfunctions. When this occurs, the vehicle operator should stop the car immediately and refer to the owner's manual to determine if the car can be driven or if it should be towed to a service station. Continued operation of the vehicle could result in damage to the engine or emissions control components, specifically the catalytic converter, a very costly component.

Sometimes the malfunction indicator light goes out by itself. This indicates that the problem that initially triggered the light no longer exists. This could happen if, for example, the gas cap was not on tight, but was then fixed. In this case, the light should reset itself and go out after several trips, eliminating the need for a service visit.

1.2 Telematics

Telematics as an interdisciplinary field that encompasses telecommunications, vehicular technologies, road transportation, road safety, electrical engineering (sensors, instrumentation, wireless communications, etc.), and computer science (multimedia, Internet, etc.). of telematics can involve any of the following:

- the technology of sending, receiving and storing information via telecommunication devices in conjunction with affecting control on remote objects
- the integrated use of telecommunications and informatics for application in vehicles and with control of vehicles on the move GNSS technology integrated with computers and mobile communications technology in automotive navigation systems (most narrowly) the use of such systems within road vehicles, also called vehicle telematics.
- In contrast, telemetry involves the transmission of measurements from the location of origin to the location of computing and consumption, especially without affecting control on the remote objects. Though typically applied in the testing of flight objects, telemetry has multiple other uses.

The main components of a Telematics System are as follows:

- Telematics Control Unit – The TCU is the embedded in-vehcile control unit that communicates with the automobile ECUs and GPS satellite and accesses the telematics services over the wireless infrastructure.
- Telematics Network Operations System – The TNOS is the hub of the operations from where all the telematics services are delivered and all the raw data from the TCUs is processed. TNOS also performs the fault management, configuration, accounting, and security functions in the telematics system.

- Wireless Communications Infrastructure – The WCI provides the backbone for all the information exchange between the TNOS and TCUs and also between the TCUs in the form of ad-hoc networks.

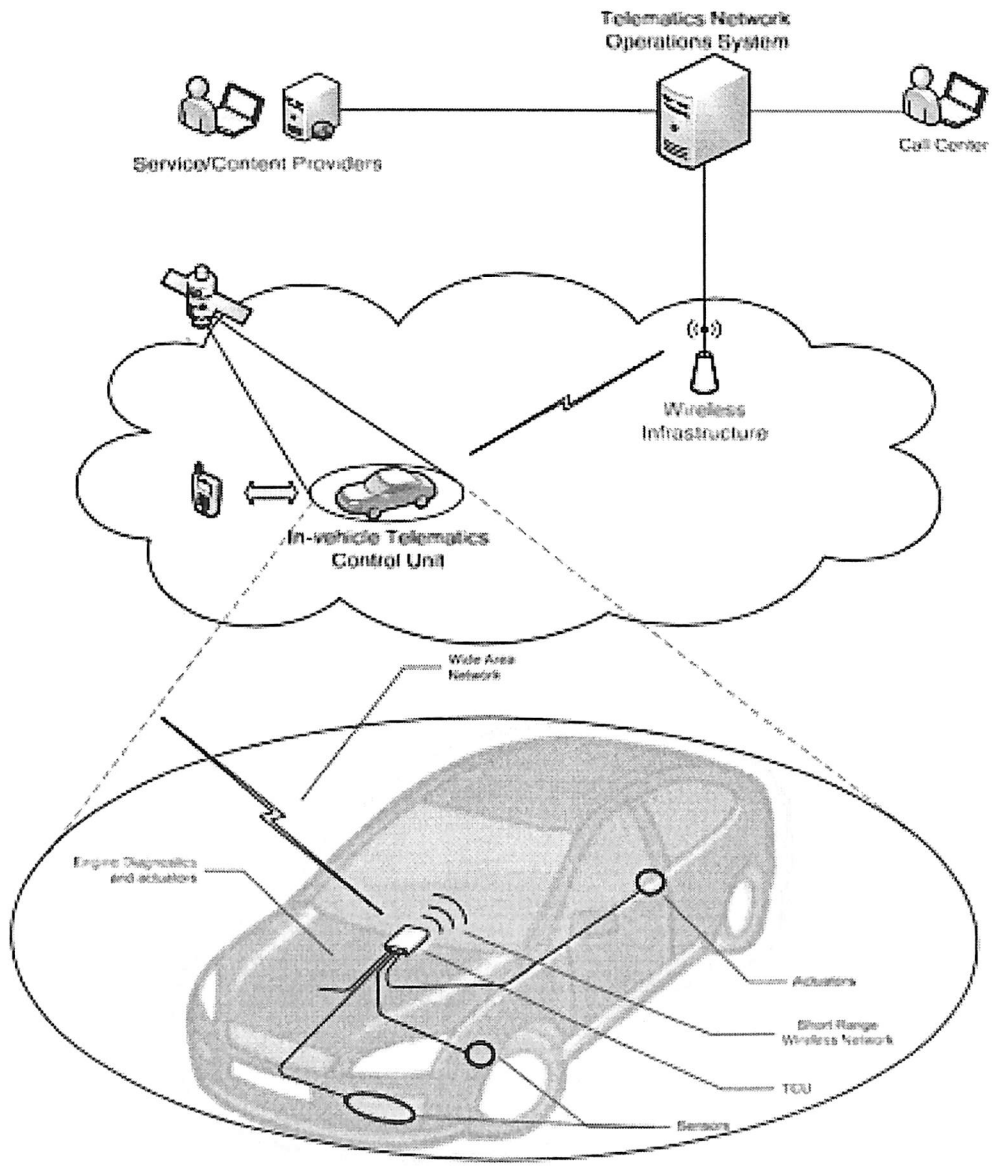


Figure 2: Vehicle Telematics Working Model

- Service Provider Call Center – The SPCC houses the customer service representatives that communicate with the vehicle occupants to provide the

emergency and non- emergency call services and access the customer and vehicle information from the TNOS.

- Service/Content Provider – The SCPs provide content such as Traffic feeds, music, video, on-demand streaming data etc to the TNOS for different telematics services.

1.3 Usage-Based Insurance

Programs like Pay-As-You-Drive (PAYD) and Pay-How-You-Drive (PHYD) enable insurers to achieve correct and deep insights into individual policyholder's driving patterns (e.g. miles driven, time of the day, variety of times the driver braked exhausting, etc.). These insights not solely modify the insurance firm to reward smart drivers with higher rates and afterwards lower claim costs however conjointly customize plans at the individual level. With an enterprise information hub (EDH), insurers will store and analyze all driving data captured through sensors. Moreover, telematics-based insurance (UBI) supply many upsides to insurers together with reduced claim prices, higher risk rating, mitigating adverse choice and financial loss, modifying risk behavior and rising complete recognition and loyalty.

1.4 Machine Learning

Machine learning systems automatically learn programs from data. This is often a very attractive alternative to manually constructing them, and in the last decade the use of machine learning has spread rapidly throughout computer science and beyond. Machine learning is used in Web search, spam filters, recommender systems, ad placement, credit scoring, fraud detection, stock trading, drug design, and many other applications. A recent report from the McKinsey Global Institute asserts that machine learning (a.k.a. data mining or predictive analytics) will be the driver of the next big wave of innovation [1]. Several fine textbooks are available to interested practitioners and researchers [2] [3]. However, much of the "folk knowledge" that is needed to successfully develop machine learning applications is not readily avail- able in them. As a result, many machine learning projects take much longer than necessary or wind up producing less- than-ideal results. Yet much of this folk knowledge is fairly easy to communicate. This is the purpose of this article.

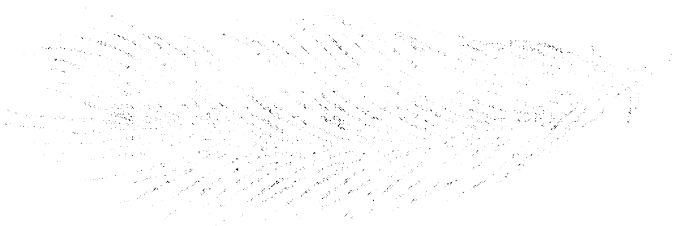
Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Table 2: The three components of learning algorithms.

1.5 Program Evaluation & Review Technique (PERT) Chart

The program (or project) evaluation and review technique, commonly abbreviated PERT, is a statistical tool, used in project management, which was designed to analyze and represent the tasks involved in completing a given project. First developed by the United States Navy in the 1950s, it is commonly used in conjunction with the *Critical Path Method (CPM)*.

CHAPTER 2.
SYSTEM ANALYSIS



2. System Analysis

2.1 Literature Review

The anomaly detection downside has been wide studied within the part and electronic merchandise fault identification community. Yu et al. have mentioned a applied math and formal logic primarily based approach to sight anomalies in GE industrial craft engines [1]. The authors analyzed statistic performance parameters to spot the foremost distinguished downside shift to pick the dataset before and once the shift. The parameter shifts were evaluated against a pre-defined set of fuzzy membership functions that capture the expected parameter shift for a particular failure.

In part community, Schimert has planned dimension reduction techniques to sight anomalies in variable statistic sensory information of the Boeing aircrafts [2]. Schimert utilized information-driven techniques principal part analysis (PCA) and freelance part analysis (ICA) to summarize the variable information into parts that specify the variance within the traditional operational data. exploitation these parts, he detected the faults by sticking out the new observation onto the parts learned exploitation traditional operational information. Associate in Nursing alert happens if the Mahalanobis distance between the new observation and PCA or ICA model is on the far side the ninety fifth or ninety-nine confidence bounds. Next, he utilized a contribution plot to research the changes within the contributions of parameters before and once the alert. The anomaly sightion downside has been additionally studied to detect anomalies sensory information of spacecrafts. In [3], Yairi et al. utilized regionally stationary autoregressive model and spectral ordering methodology to sight the amendment points in statistic mensuration information. The results were incontestable exploitation many visual image techniques.

The anomaly detection is additionally a number one concern for natural philosophy makers. archangel et al. [4] have applied Mahalanobis distance (MD) primarily based approach to

sight anomalies in electronic merchandise. The authors learned baseline MD distribution of healthy merchandise by normalizing the information and computing MD values. Then, the check information was normalized with the mean and variance (SD) of the baseline. The MD values of the check information was computed exploitation the matrix from the baseline information. To sight the anomalies, the MD values of the check information was compared with the baseline MD values. If the check information has MD values outside 3SD then it indicated the abnormal behavior. The authors incontestable the approach on electronic merchandise information obtained exploitation experiments.

There are recent efforts on observation health of fleet vehicles. In [5] Saxena et al. have planned a dynamic case-based reasoning framework to take care of health of crucial systems. Author developed a Natural Language Processing (NLP) technique to extract info from the text Associate in Nursingd incontestable the technique on an automobile dataset. The data-driven fault detection and identification has been additionally planned for cars. Namburu et al. [6] planned a scientific data-driven framework to sight and diagnose faults in automotive engines. They applied the data-driven framework to Associate in Nursing experimental system consisting of a Toyota Camry engine running with manual transmission on a measuring system check stand. The automotive field failure information has been additionally analyzed in our previous work [7] wherever we tend to utilized dimension reduction and bunch techniques to sight anomalies in DTCs. during this paper, we tend to specialise in detection anomalies in repair and failure information of cars. Our objective is to sight anomalies and report back to quality and repair engineers in order that they'll take applicable measures.

2.2 Traditional Analytics Approach

Traditionally the majority of business intelligence and analytics solutions are designed round the construct of batch operations that move data between totally different persisted knowledge stores, like relative databases or analytics cubes. Users would then issue a question against the data at rest to support situations like ad-hoc analysis, dashboards or scorecards. whereas this approach has tested to be extremely victorious for variety of years, and still remains an extremely relevant resolution to business operations nowadays, the sheer volume and rate of data being generated by trendy applications or devices is putting pressure on this existing paradigm. the following diagram illustrates this approach.

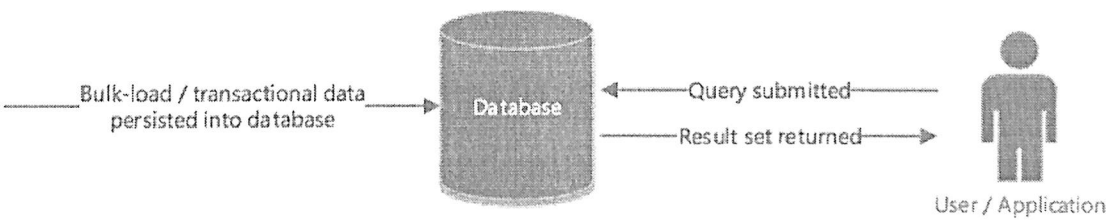


Figure 3: Traditional Data Analytics

To circumvent a number of the challenges associated with timeliness of data being made available to DSS, companies have often looked to optimize their architectures using techniques like accelerating data loading schedules, or manually scaling out data transformation processes. This has proven to be a successful remediation for faster Data Availability, however data availability is generally limited to minutes at best. There is a limit that an extract / load process can be completed and instantiated against source system being restarted.

2.3 Motivation

Conventionally, anomaly detection and identification strategies supported Apriori skilled knowledge and synthesis method like skilled systems and model-based reasoning are mainly studied for this purpose. whereas these knowledge-intensive approaches are tried to perform far better than the classical limit checking technique, it's usually pricey and long to arrange the knowledge-base or model-base that square measure needed for them.

On the opposite hand, in recent years, generalization techniques thus known as Data-Mining (DM) or Machine Learning (ML) technologies have drawn a lot of attention as various approaches to the anomaly detection issues in numerous application fields. Development purpose have thought-about the potential worth of the vehicle measure information hold on in ground stations, and studied DM/ML-based anomaly detection and fault identification strategies for time period identification systems.

2.4 Objective

Telemetry data is that the solely supply for identifying/predicting anomalies in vehicle. Human specialist analyze this data in real time, however its massive volume, makes this analysis very tough. within the project we studied varied clump and anomaly detection algorithmic program to assist organization playacting mensuration analysis. Real cases of auto anomalies square measure considered, permitting assessing the effectiveness of algorithmic program (Principal component Analysis), that showed to be effective within the case study where several telemetry channels tended to deliver outlier values and, in these cases, every record may be thought-about as some extent in a very 3-D area outlined by its tier pressure, Engine Oil, and Engine Temperature coordinates. To capture this outliers, we are able to project original information in 2-Dimensional area victimisation PCA. This parameter plays a very important role in applying PCA-based anomaly detection.

The solution is enforced as a lambda architecture pattern showing the total potential for real-time and batch processing. It includes a Vehicle Telematics simulator, leverages Event Hubs for ingesting several simulated vehicle mensuration events into Azure, then uses Stream Analytics for gaining period insights on vehicle health and persists that information into long storage for batch analytics. Machine Learning a bonus for anomaly detection in period and batch processing to realize predictive insights.

2.5 Methodology Overview

Implementation proposes a unique driving behavior analysis technique supported the vehicle OBD info and PCA formula. the strategy proposed collects the vehicle operational data, comprehensive vehicle speed, odometer, outside worker, within worker, tire pressure, ABS, throttle position, engine RPM, Engine load, vehicle build, general service info.

Our goal here is to predict the vehicles that need maintenance or recall supported sure heath statistics. Example were the subsequent assumptions area unit considered:

Vehicles need service maintenance if (three conditions == TRUE):

- ✓ Tire pressure is low
- ✓ Engine oil level is low
- ✓ Engine temperature is high

Vehicles might have a security issue and need recall if (one condition == TRUE):

- ✓ Engine temperature is high however outside temperature is low
- ✓ Engine temperature is low however outside temperature is high

2.6 System Requirements: (Software/Hardware):

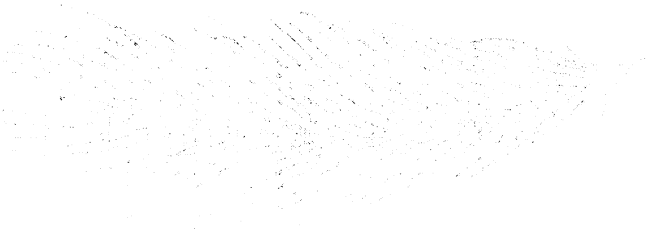
2.6.1 Development Environment:

1. Visual Studio 2015 SP3 or Higher.
2. Microsoft Azure. (PAAS)
3. Enterprise - Microsoft PowerBI.

2.6.2 User Environment:

1. OBD data Transmitter,
2. Smart Phone (Android 4.0 +, Windows, iOS)
3. Data Connectivity

CHAPTER 3.
SYSTEM DESIGN



3.System Design

3.1 Usecase Diagrams

3.1.1 Data flow Diagram

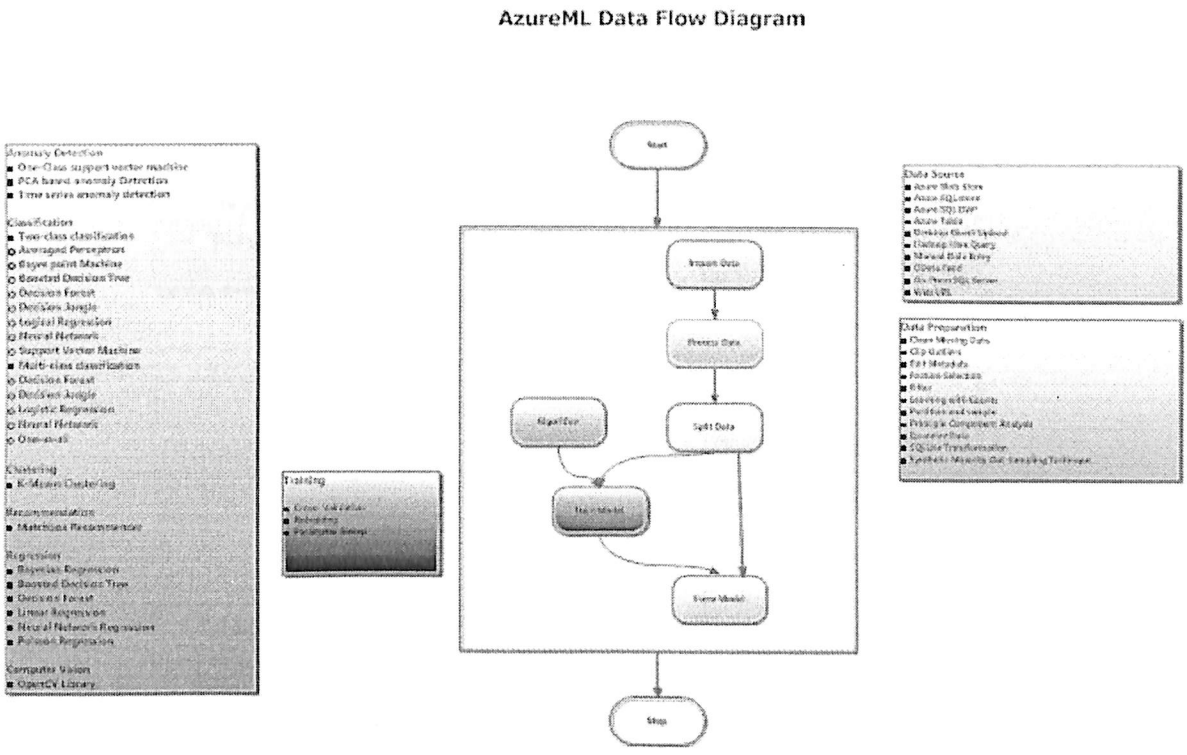


Figure 4: Data Flow Diagram Level 0

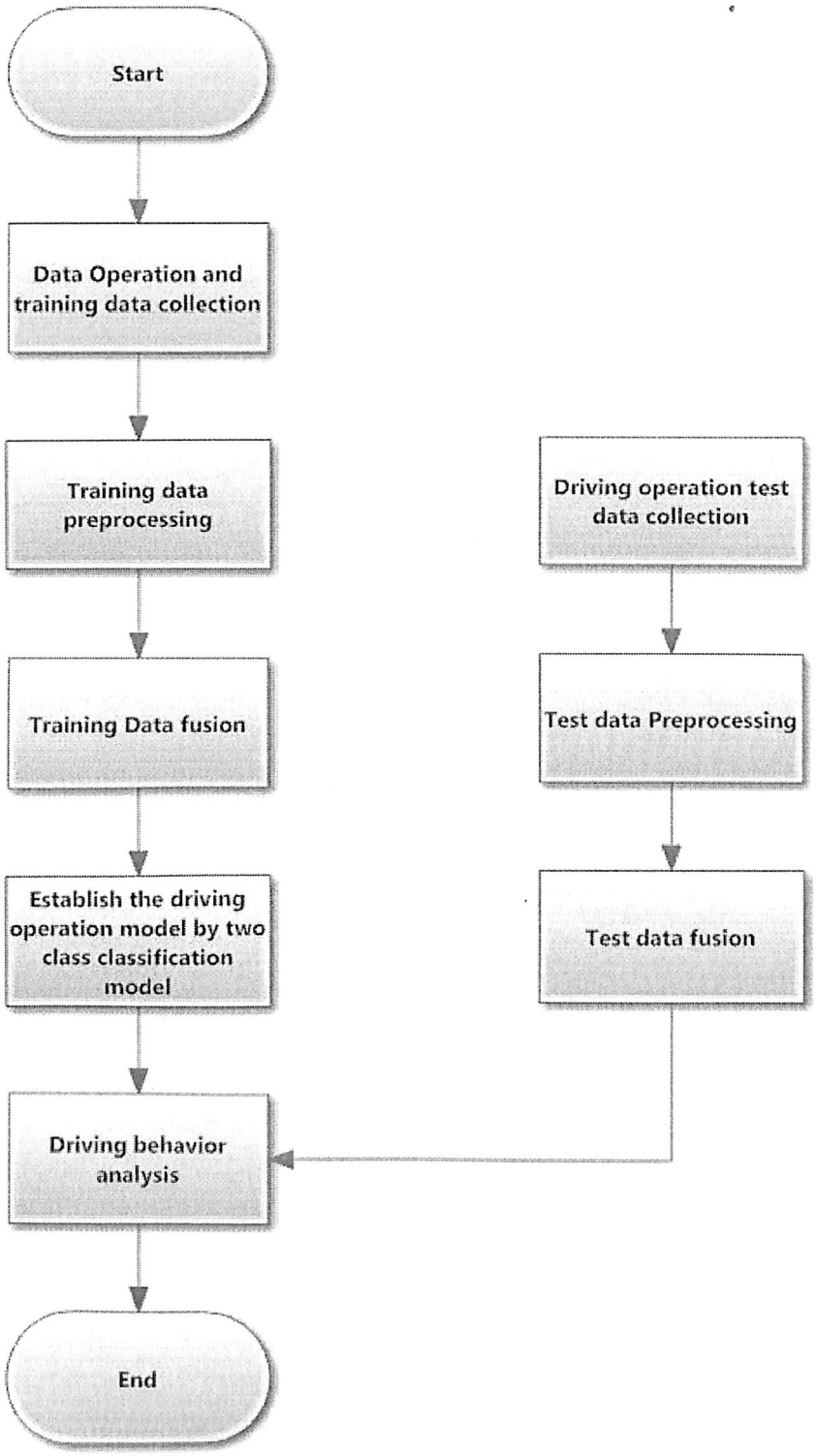


Figure 5: Data Flow Diagram Level 1

3.1.2 Sequence Diagram

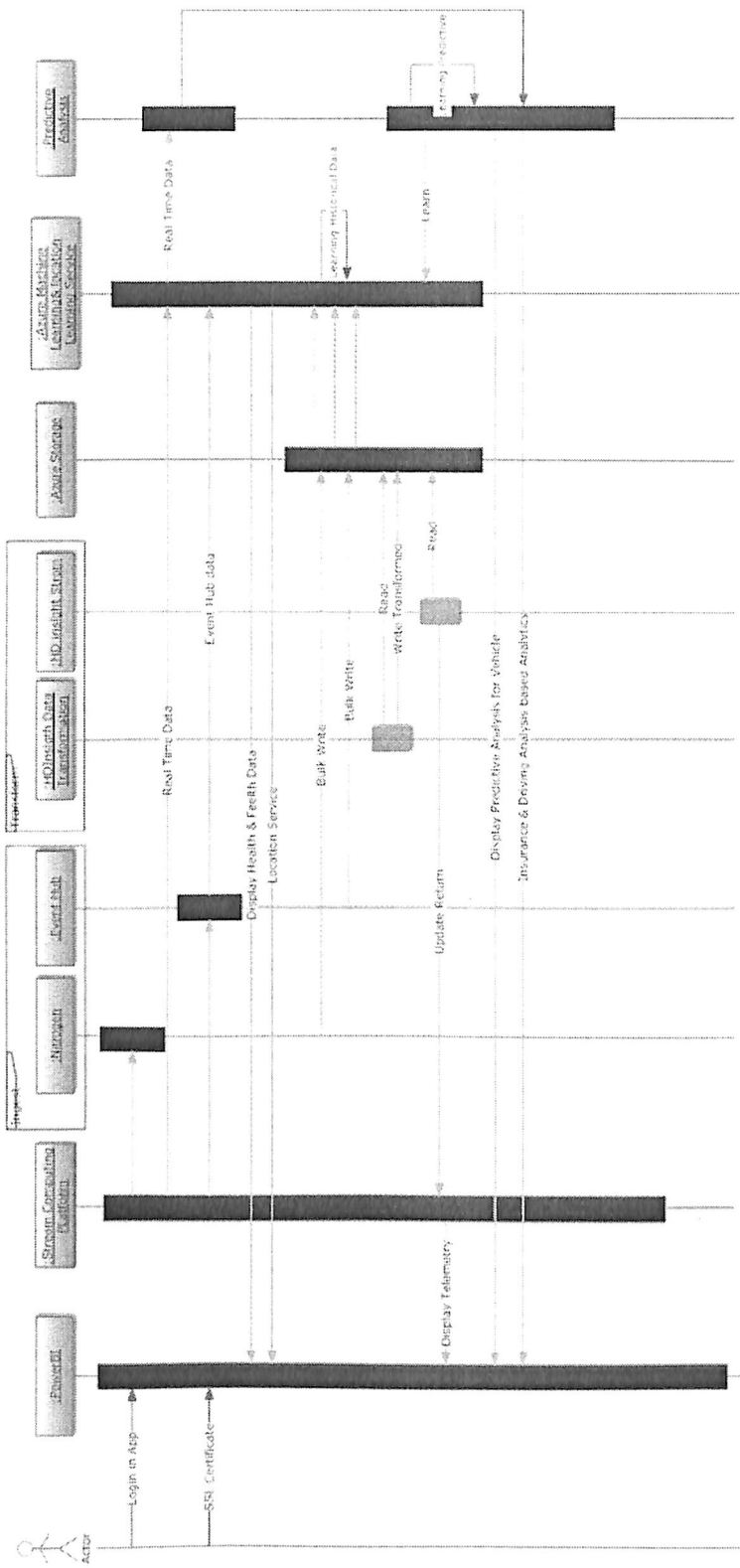


Figure 6: Sequence Execution Diagram

3.1.3 Component Diagram

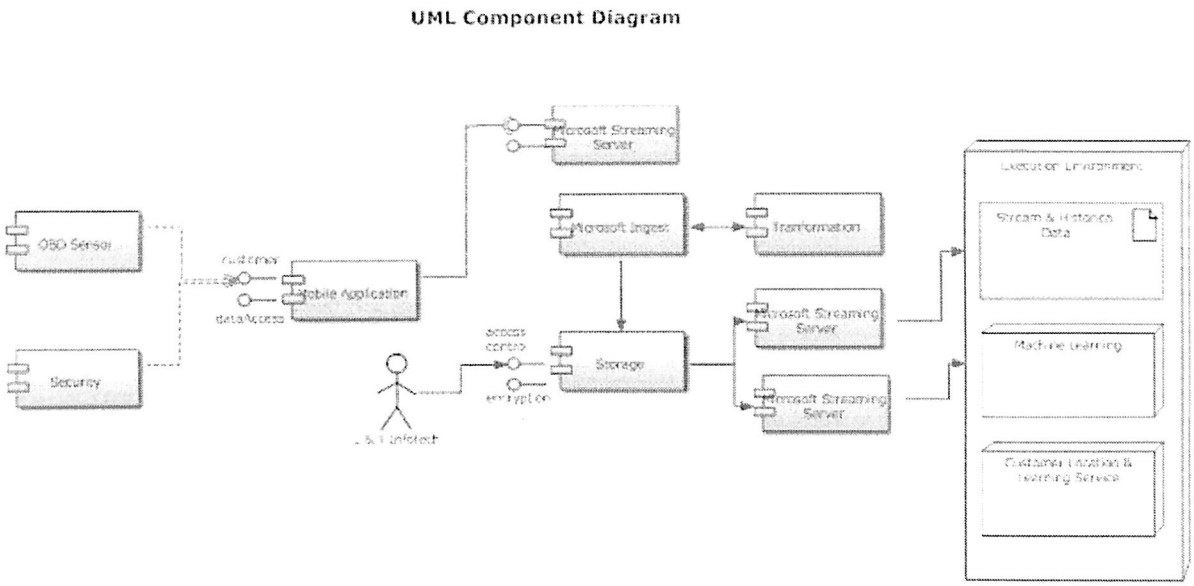


Figure 7: Component Diagram

3.1.4 Deployment Diagram

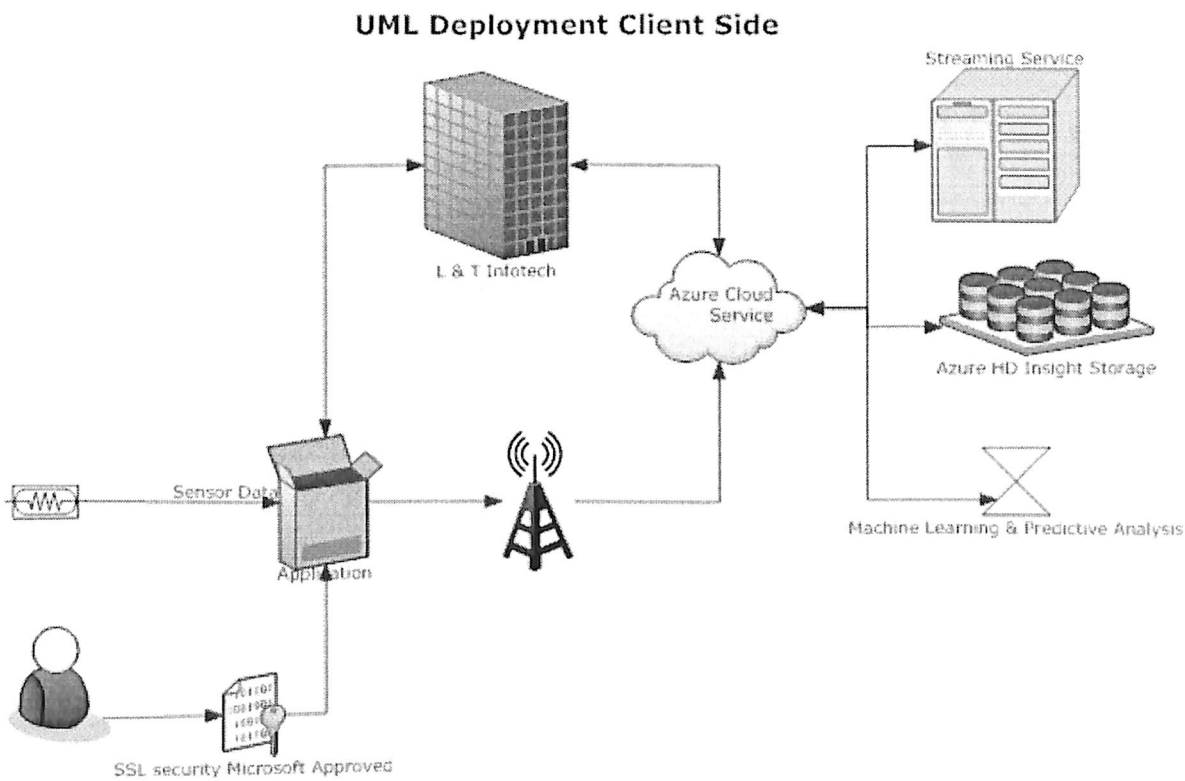
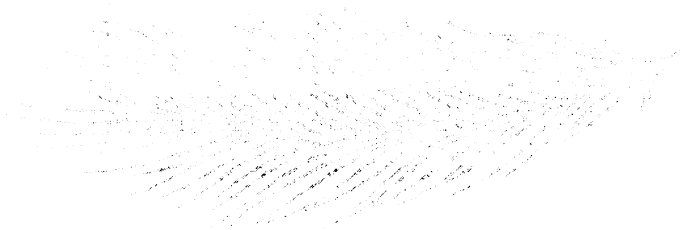


Figure 8: Deployment Diagram

CHAPTER 4.
PROJECT OVERVIEW



4. Project Overview

Super computers have moved out of the lab and are now parked in our garage! These cutting-edge automobiles contain a myriad of sensors, giving them the ability to track and monitor millions of events every second. We expect that by 2020, most of these cars will be connected to the internet. Imagine tapping into this wealth of data to provide best in class safety, reliability and driving experience. Microsoft has made this imagination a reality via Cortana Analytics.

Microsoft's Cortana Analytics is a fully managed big data and advanced analytics suite that enables you to transform your data into intelligent action. We want to introduce you to the Cortana Analytics Vehicle Telemetry Analytics Solution Template. This solution demonstrates how car dealerships, automobile manufacturers and insurance companies can use the capabilities of Cortana Analytics to gain real-time and predictive insights on vehicle health and driving habits.

The solution is implemented as a lambda architecture pattern showing the full potential of the Cortana Analytics platform for real-time and batch processing. It includes a Vehicle Telematics simulator, leverages Event Hubs for ingesting millions of simulated vehicle telemetry events into Azure, then uses Stream Analytics for gaining real-time insights on vehicle health and persists that data into long-term storage for richer batch analytics. It takes advantage of Machine Learning for anomaly detection in real-time and batch processing to gain predictive insights. HDInsight is leveraged to transform data at scale, and Data Factory handles orchestration, scheduling, resource management and monitoring of the batch processing pipeline. Finally, Power BI gives this solution a rich dashboard for real-time data and predictive analytics visualizations.

4.1 Architecture

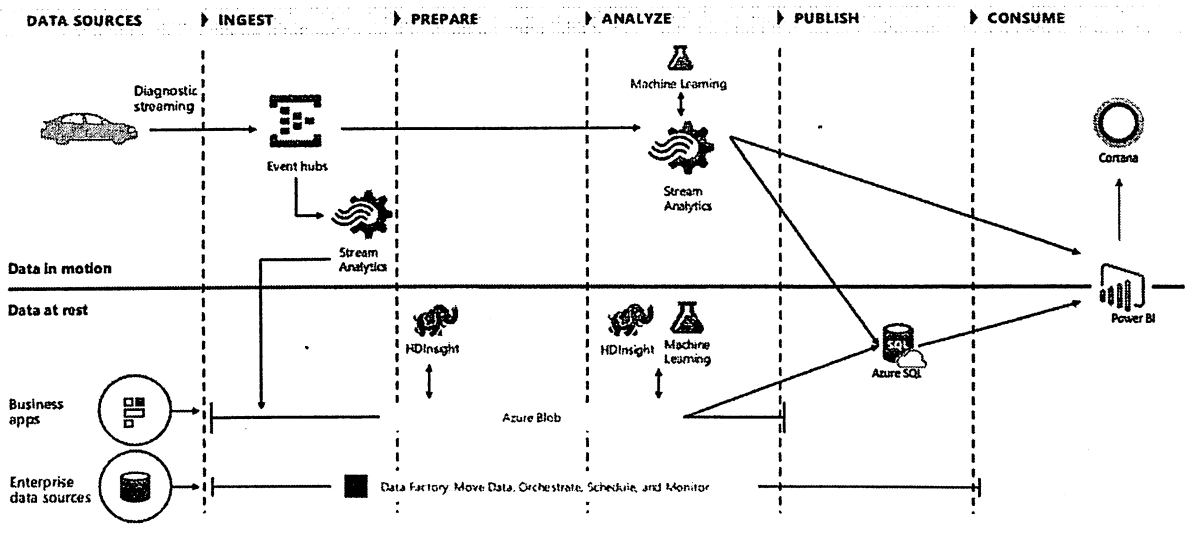


Figure 9: Vehicle Telemetry Analytics Solution Architecture

This solution includes the following **components** and showcases their end to end integration,

- **Event Hubs** for ingesting millions of vehicle telemetry events into Azure.
- **Stream Analytics** for gaining real-time insights on vehicle health and persists that data into long-term storage for richer batch analytics.
- **Machine Learning** for anomaly detection in real-time and batch processing to gain predictive insights.
- **HDInsight** is leveraged to transform data at scale
- **Data Factory** handles orchestration, scheduling, resource management and monitoring of the batch processing pipeline.
- **Power BI** gives this solution a rich dashboard for real-time data and predictive analytics visualizations.
- This solution accesses two different **data sources**:

- **Simulated vehicle signals and diagnostics**: A vehicle telematics simulator emits diagnostic information and signals that correspond to the state of the vehicle and the driving pattern at a given point in time.
- **Vehicle catalog**: A reference dataset containing a VIN to model mapping. Technology

4.2 Azure Stream Analytics

Enabling a real-time analytics solution based on streaming data provides a solution to a number of the aforementioned challenges of real-time data at scale. The model for the implementation represents a significant shift by moving from point queries against stationary data, to a standing temporal query that consumes moving data. Fundamentally, we enable insight on the data before it is stored in the analytics repository. As such, companies gain the benefits of real-time insights on data as business events occur, but also the ability to store this information in a robust repository for historical analysis at a later time. The following diagram illustrates the approach to real-time analytics, demonstrating the contrast of Figure 3: Traditional Data Analytics to show how real-time analytics is applied.

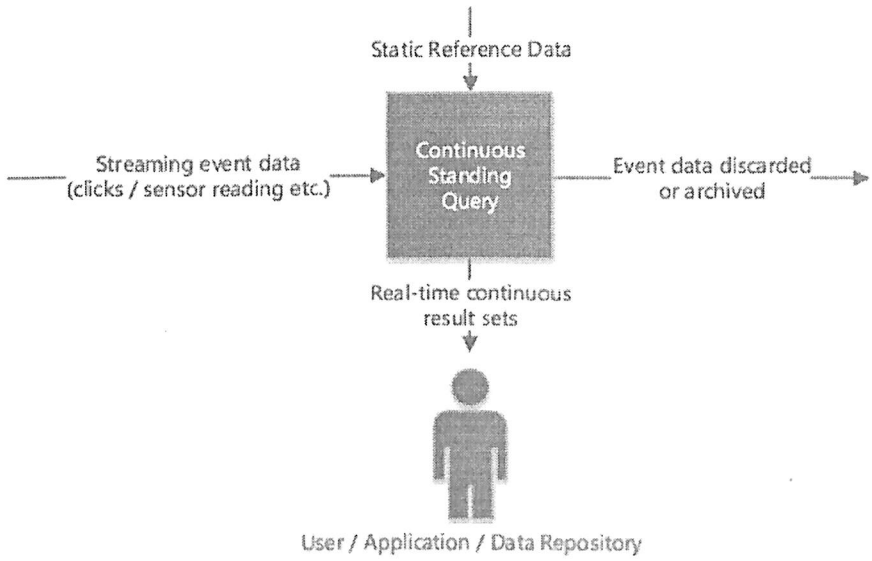
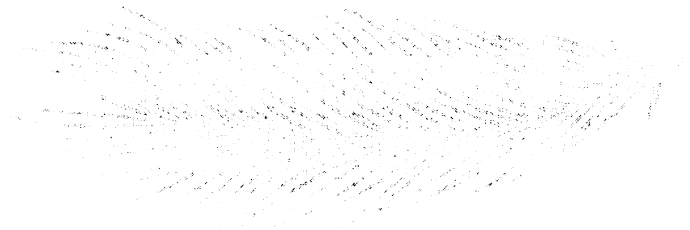


Figure 10: Stream Analytics

CHAPTER 5. IMPLEMENTATION



5. Implementation

5.1 Algorithms

5.1.1 Principle Component Algorithm

Principal component analysis (PCA) is among the foremost well-liked tools in machine learning, statistics, and information analysis additional typically. PCA is that the basis of the many techniques in data mining and information retrieval, as well as the latent linguistics analysis of enormous databases of text and hypertext mark-up language documents delineated in [3] during this paper, we have a tendency to figure PCAs of terribly giant information sets via a randomised version of the block Lanczos technique, summarized in Section a pair of below. The proofs in [4] show that this technique needs solely a handful of iterations to provide nearly best accuracy, with irresistibly high likelihood (the likelihood is in- dependent of the information being analyzed, and is usually $1 - 10^{-15}$ or greater). The randomised formula has several benefits, as shown in [3] [4]; the current arti- cle adapts the formula to be used with information sets that area unit large to be hold on within the RAM of a typical computing system.

Computing a PCA of a data set amounts to constructing a singular value decomposition (SVD) that accurately approximates the matrix A containing the information being analyzed (possibly when appropriately "normalizing" A, for instance, by subtracting from each column its mean). That is, if A is m × n, then we have a tendency to realize a positive whole number k, min(m, n) and construct matrices U , Σ, and V specified

$$A \approx U \Sigma V^T$$

with U being an m × k matrix whose columns are orthonormal, V being an n × k matrix whose columns are orthonormal, and Σ being a diagonal k × k matrix whose entries are all nonnegative. Most often, the relevant measure of the quality of the approximation.

$$A - U \Sigma V^T$$

PCA Algorithm Following six steps:

6. Using a random number generator, form a real $n \times l$ matrix G whose entries are independent, identically distributed Gaussian random variables of zero mean and unit variance , and compute the $m \times l$ matrices H^0, H^1, H^{l-1}, H^i defines via the formula:

$$H^0 = AG$$

$$H^1 = A^T G^o$$

$$H^2 = A^T G^1$$

Form the $H = H^0 \mid H^1 \mid H^2 \ldots\ldots\ldots\ldots \mid H^i$.

7. Using a pivoted QR-decomposition, form a real $(M\ X\ (I + 1)l)$ matrix Q columns are orthonormal, such that there exists a real ((I + 1)l) × ((I + 1)l) matrix R for which

$$H = Q\ R$$

8. Compute the $n \times ((I\ +\ 1)l)$ Product matrix,

$$T = A^T Q$$

9. From SVD of T,

$$T = \hat{V}\ X\ W^T$$

10. Compute the $m \times ((I\ +\ 1)l)$ product matrix,

$$U = Q\ X\ W$$

11. Retrieve the leftmost m×k block U of $\hat{U}$, the leftmost $N\ X\ K$ block V of $\hat{V}$, and the leftmost uppermost K X K blocks Σ of Σ. The product $U \Sigma V^T$ than approximates.

5.1.2 Anomaly Detection

Anomaly Detection API is an example built with Azure Machine Learning that detects anomalies in time series data with numerical values that are uniformly spaced in time.

This anomaly detection service can detect the following different types of anomalies on time series data:

- Positive and negative trends: When monitoring memory usage in computing, for instance, an upward trend is indicative of a memory leak,
- Increase in the dynamic range of values: As an example, when monitoring the exceptions thrown by a service, any increases in the dynamic range of values could indicate instability in the health of the service, and
- Spikes and Dips: For instance, when monitoring the number of login failures to a service or number of checkouts in an e-commerce site, spikes or dips could indicate abnormal behavior.

These detectors track changes in values over time and reports ongoing changes in their values. They do not require adhoc threshold tuning and their scores can be used to control false positive rate. The anomaly detection API is useful in several scenarios like service monitoring by tracking KPIs over time, usage metrics such as number of searches, numbers of clicks, performance counters like memory, cpu, file reads, etc. over time.

The underlying principle of any statistical anomaly detection technique is: "An anomaly is an observation which is suspected of being partially or wholly irrele- vant because it is not generated by the stochastic model assumed" [Anscombe and Guttman 1960]. Statistical anomaly detection techniques are based on the following key assumption:

Assumption: "*Normal data instances occur in high probability region so fast orchestic model, while anomalies occur in the low probability regions of the stochastic model.*"

5.1.2.1 Parametric Technique

As mentioned before, parametric techniques assume that the normal data is generated by a parametric distribution with parameters Θ and probability density function $f(x,\Theta)$, where x is an observation. The anomaly score of a test instance (or observation) x is the inverse of the probability density function, $f(x,\Theta)$. The parameters Θ are estimated from the given data.

Alternatively, a statistical hypothesis test (also referred to as discordancy test in statistical outlier detection literature [Barnett and Lewis 1994]) maybe used. The null Hypothesis (H) for such tests is that the data instance x has been generated using the estimated distribution (with parameters Θ). If the statistical test rejects H, x is declared to be anomaly. A statistical hypothesis test is associated with a test statistic, which can be used to obtain a probabilistic anomaly score for the data instance x. Based on the type of distribution assumed, parametric techniques can be further categorized as follows:

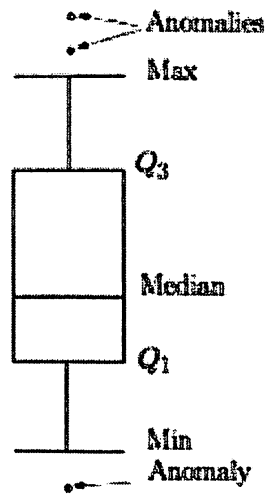


Figure 11: A box plot for a univariate data set.

Generated from a Gaussian distribution. The parameters are estimated using Maximum Likelihood Estimates (MLE). The distance of a data instance to the estimated mean is the anomaly score for that instance. A threshold is applied to the anomaly scores to determine

the anomalies. Different techniques in this category calculate the dis- tance to the mean and the threshold in different ways.

A simple outlier detection technique, often used in process quality control domain [Shewhart 1931], is to declare all data instances that are more than 3σ distance away from the distribution mean μ, where σ is the standard deviation for the distribution. The $\mu \pm 3\sigma$ region contains 99.7% of the data instances.

More sophisticated statistical tests have also been used to detect anomalies, as discussed in [Barnett and Lewis 1994; Barnett 1976; Beckman and Cook 1983]. We will describe a few tests here.

5.2 Deep Drive & Cook Book

5.2.1 Data Sources

The solution uses 2 totally different data sources:

- Simulated Vehicle signals and diagnostic dataset,
- Vehicle catalog.

A vehicle telematics simulator is enclosed as a part of this solution. It emits diagnostic data and signals akin to the state of the vehicle and driving pattern at a given purpose in time. Click Vehicle Telematics machine to transfer the Vehicle Telematics Machine Visual Studio resolution for customizations supported your needs.

The vehicle catalog contains a reference dataset with a VIN to model mapping.

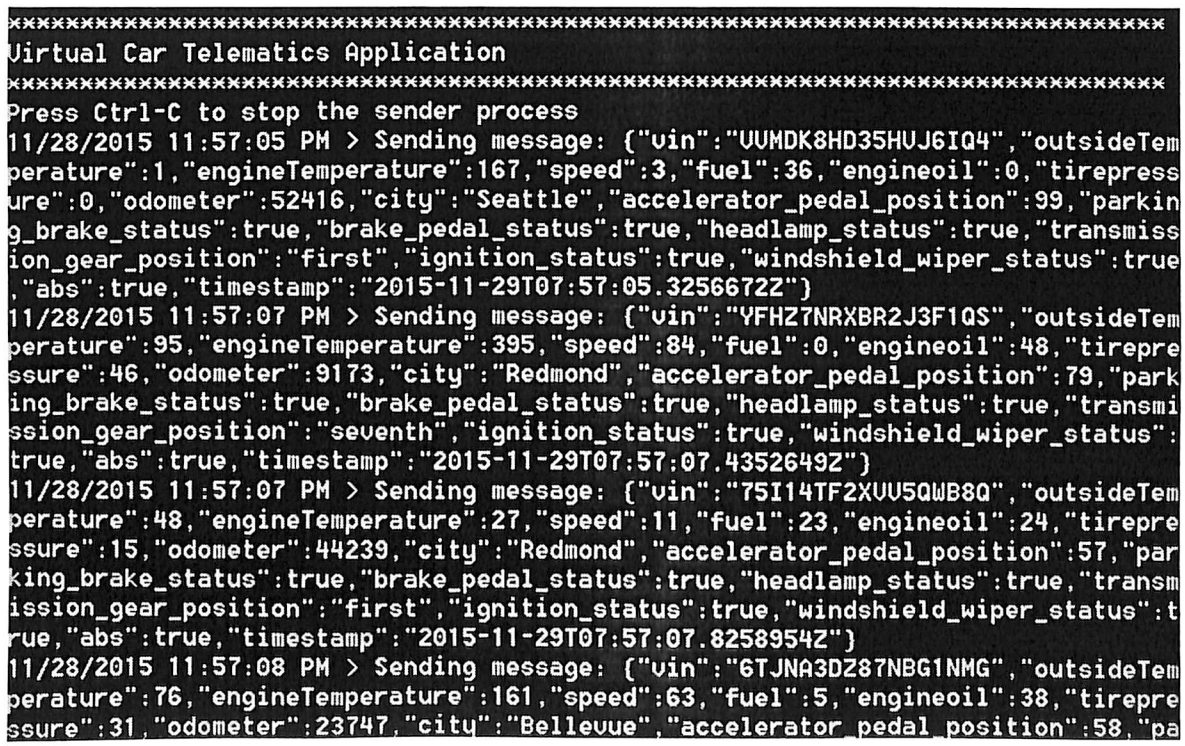


Figure 12: JSON Format

This is a JSON format dataset and contains the below schema,

To generate simulated data

1. Click on the arrow(→) on the upper right on the Vehicle Simulator node in to download the data simulator package & extract the files locally on your machine.

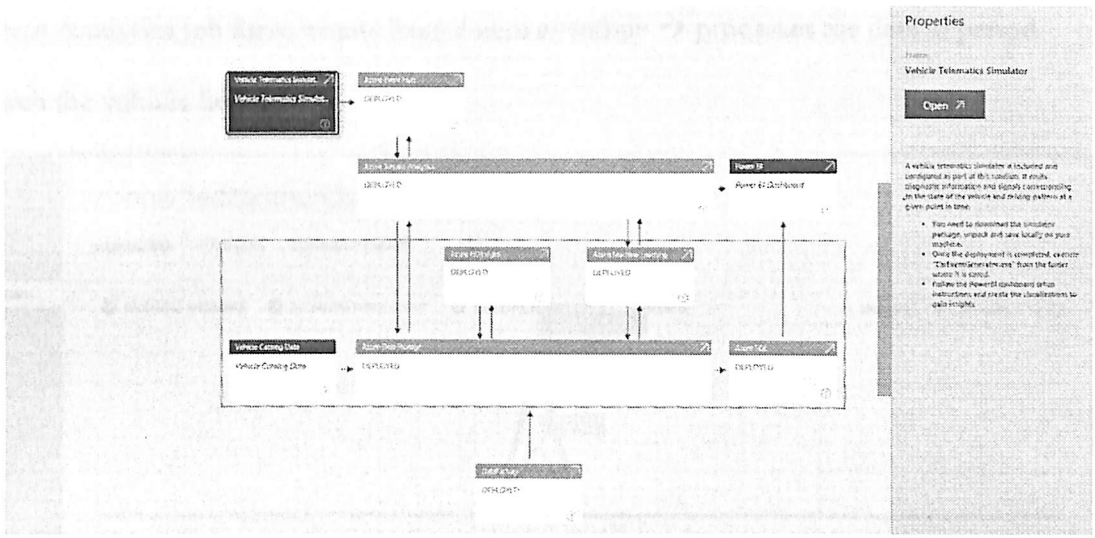


Figure 13: Vehicle Tlemetry Analytics Solution Blueprint

On your local machine, go to the folder where you extracted the Vehicle Telematics Simulator package.

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
ADF	File folder					8/26/2015 11:00 PM
ASA	File folder					8/26/2015 11:00 PM
bin	File folder					8/26/2015 11:00 PM
CarEventGenerator	File folder					8/26/2015 11:00 PM
RealTimeDashboardApp	File folder					8/26/2015 11:00 PM
referencedata	File folder					8/26/2015 11:00 PM
scripts	File folder					8/26/2015 11:00 PM

Figure 8: Vehicle Telematics Simulator folder

2. Execute the application **CarEventGenerator.exe**.

5.2.2 Real-time analysis

The JSON generated by the Vehicle Telematics simulator are printed to the Event Hub victimization the Event Hub SDK.

The stream Analytics job these events from Azure eventhub → processes the data in period to research the vehicle health.

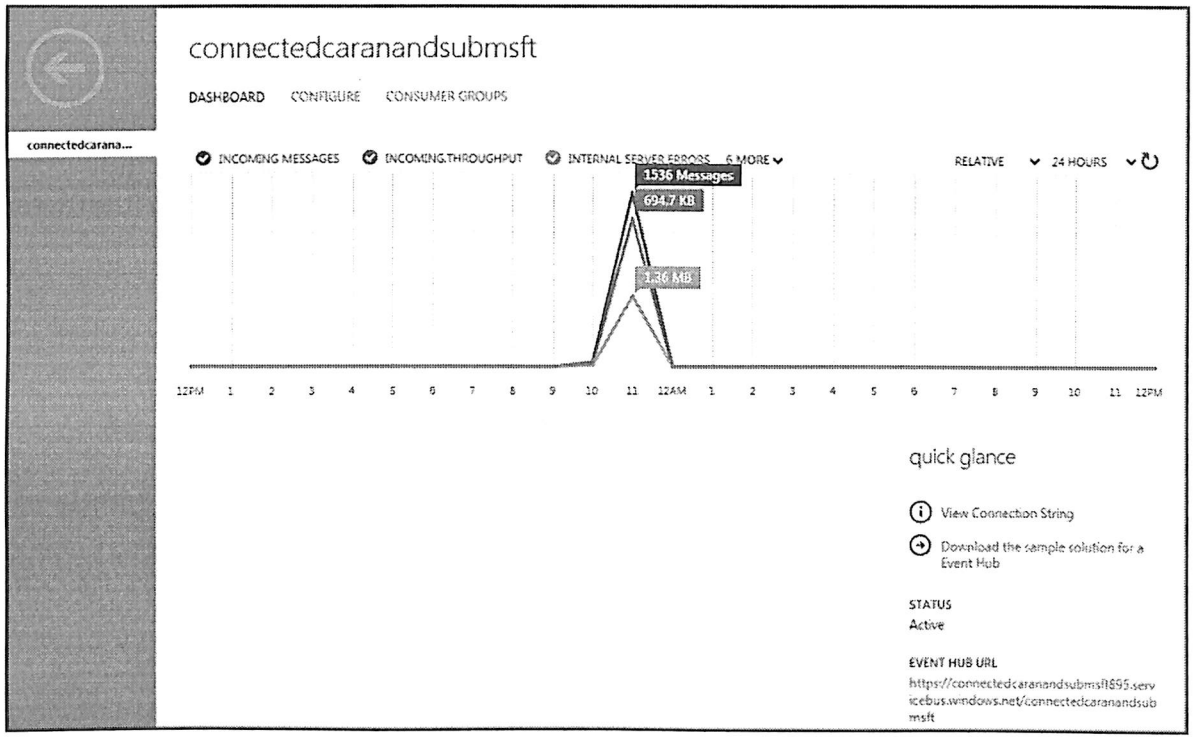


Figure 14: Dashboard:Event Hub

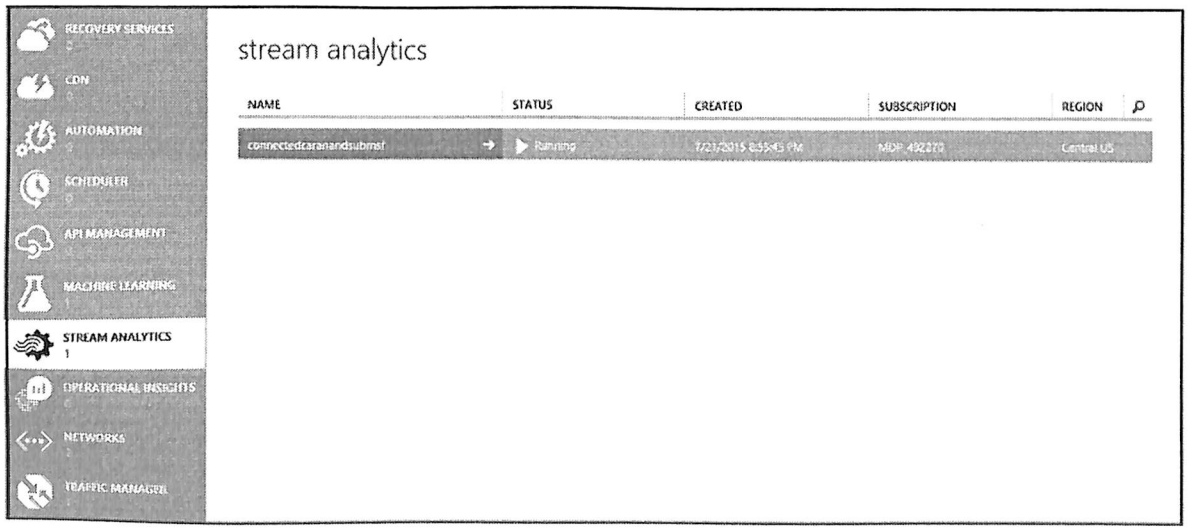


Figure 15: Stream analytics: job processing data

The stream analytics job ingests data from the Event Hub, performs a be part of with the reference data to map the vehicle VIN to the corresponding model and additionally persists them into Azure blob storage for wealthy batch analytics.

The below stream analytics query is employed to persist the data into Azure blob storage.

```
Select EventHubSource.vin, BlobSource.Model, EventHubSource.timestamp,
EventHubSource.outsideTemperature, EventHubSource.engineTemperature,
EventHubSource.speed, EventHubSource.fuel, EventHubSource.engineoil,
EventHubSource.tirepressure, EventHubSource.odometer, EventHubSource.city,
EventHubSource.accelerator_pedal_position, EventHubSource.parking_brake_status,
EventHubSource.headlamp_status, EventHubSource.brake_pedal_status,
EventHubSource.transmission_gear_position, EventHubSource.ignition_status,
EventHubSource.windshield_wiper_status, EventHubSource.abs into BlobSink from
EventHubSource join BlobSource on EventHubSource.vin = BlobSource.VIN
```

Figure 16: Stream analytics job query for data ingestion

5.2.2.1 Batch Analysis

We also are generating an extra volume of simulated vehicle signals and diagnostic dataset for richer batch analytics. this can be needed to confirm an honest representative data volume for batch processing. For this purpose, we have a tendency to are employing a pipeline named 'PrepareSampleDataPipeline' within the Azure knowledge plant progress to get annual price of simulated vehicle signals and diagnostic dataset.

Click data plant custom activity to transfer the data → custom DotNet activity .

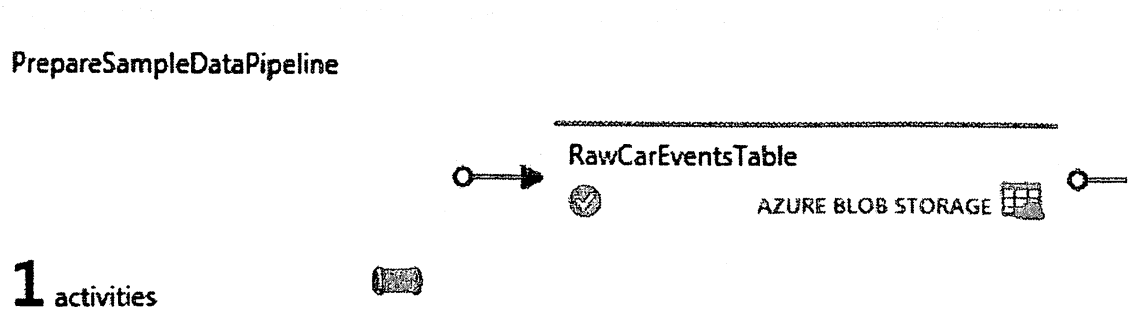


Figure 17: Sample data: batch processing workflow

The pipeline consists of a custom ADF .Net Activity, show below,

Figure 18: Data Pipeline

Once the pipeline executes with success and 'RawCarEventsTable' dataset is marked 'Ready', annual price of simulated vehicle signals and diagnostic data ar created. you'll see the subsequent folder and file created in your storage account beneath the 'connectedcar'.

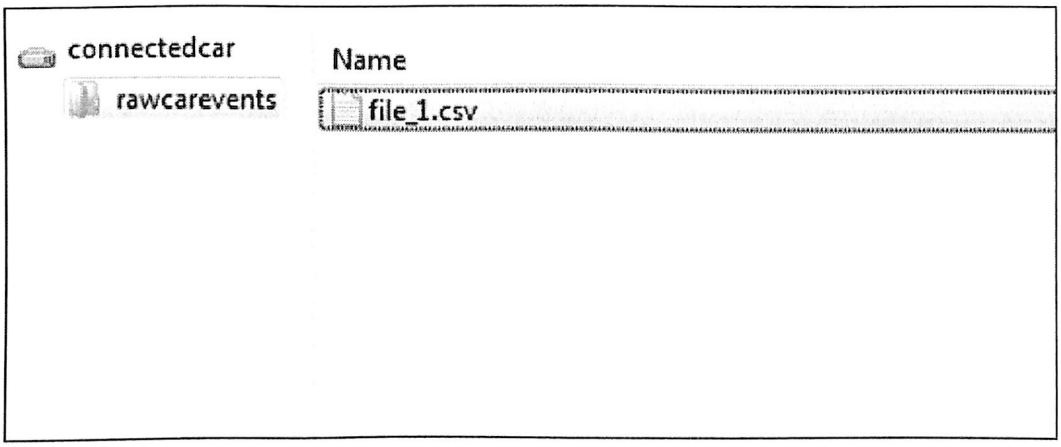


Figure 19: SampleDataPipeline: Output

5.2.3 Prepare

The raw semi-structured vehicle signals and diagnostic dataset is partitioned off within the data preparation step into a YEAR/MONTH format for economical querying and climbable long term storage (i.e. it allows faulting over from one blob account to consequent because the initial fills up).

The output data (labeled PartitionedCarEventsTable) is to be unbroken for a long amount because the foundational /"rawest" kind of data within the customer's "Data Lake".

The {input knowledge|input file|computer file} to the present pipeline would usually be discarded because the output data has full fidelity to the input – its simply hold on {Partitioned} higher for resulting use

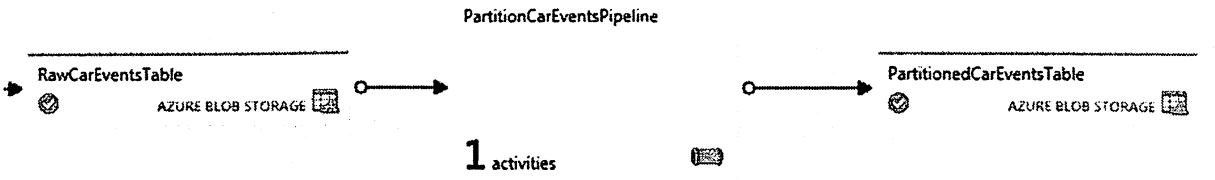


Figure 20: Partition Car Events workflow

The raw data is partitioned using a Hive HDInsight activity in 'PartitionCarEventsPipeline'. A annual cost of sample data generated in step 1 is partitioned by YEAR/MONTH (total = 12 partitions) in a year.

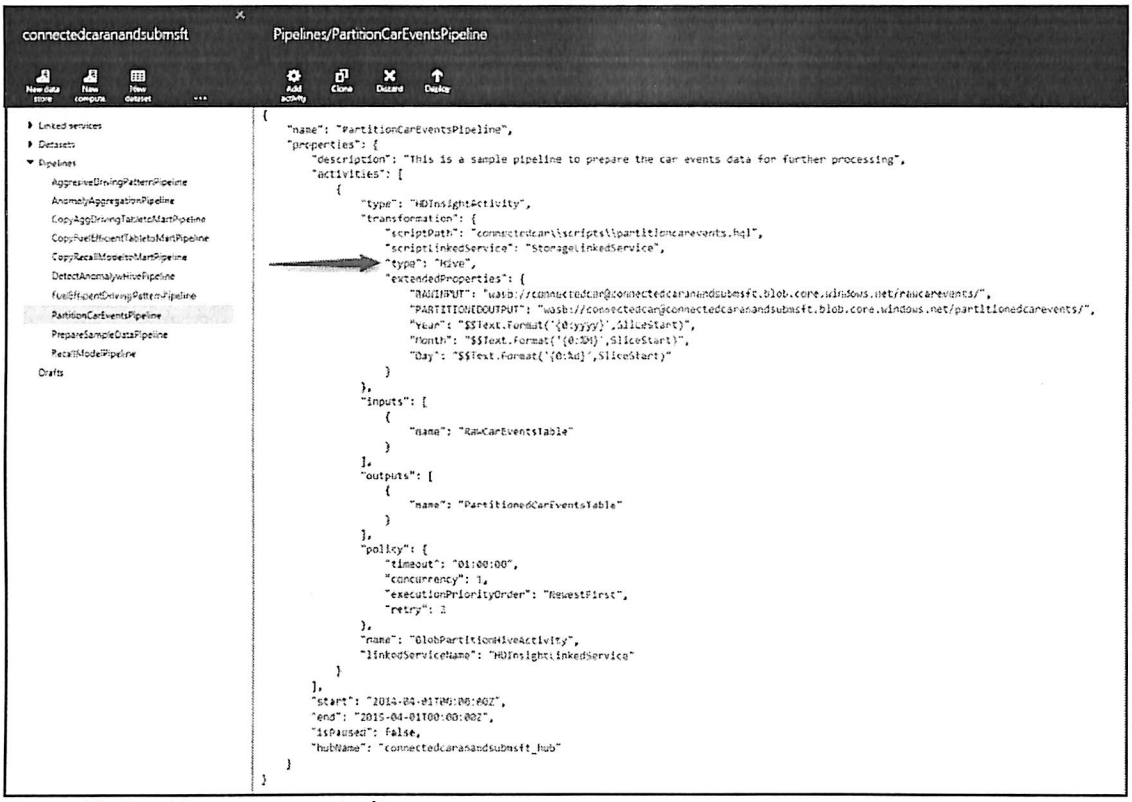


Figure 21: Partition car event pipeline

The Hive script shown below,

named 'partitioncarvents.hql → is used for partitioning

location → '\src\connectedcar\scripts'.

```
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode = nonstrict;
set hive.cli.print.header=true;

DROP TABLE IF EXISTS RawCarEvents;
CREATE EXTERNAL TABLE RawCarEvents
(
        vin                             string,
        model                           string,
        timestamp                       string,
        outsidetemperature              string,
```

```
enginetemperature            string,
speed                        string,
fuel                         string,
engineoil                    string,
tirepressure                 string,
odometer                     string,
city                         string,
accelerator_pedal_position   string,
parking_brake_status         string,
headlamp_status              string,
brake_pedal_status           string,
transmission_gear_position   string,
ignition_status              string,
windshield_wiper_status      string,
abs                          string,
gendate                      string
)
```

Once the pipeline is executed successfully, following partitions generated in your account under the 'connectedcar' storage container.

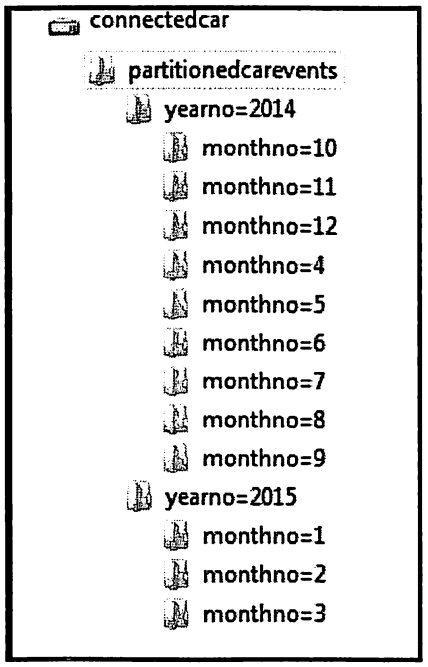


Figure 22: Partitioned Output

5.3 Data Analysis

The combination of Azure Machine Learning, Azure Data Factory & Azure Stream Analytics Finally Azure HDInsight for rich advanced analytics on driving habits & Vehicle Health. There are 3 sub-sections here:

1. **Machine Learning**: contains information on the anomaly detection experiment that we have used in this solution to predicting vehicle servicing & vehicles requiring recalls due to safety issues
2. **Real-time analysis**: This contains data regarding the real-time analytics using the Stream Analytics Query (SAQ) Language and the machine learning experiment in real-time using a application
3. **Batch analysis**: This contains data regarding the processing and transformation of the batch data using Azure HDInsight.

5.3.1 Machine Learning

Our goal here is to predict the vehicles that require maintenance or recall based on certain heath statistics. We make the following assumptions

Vehicles require **servicing maintenance** if one of the following three conditions are true:

- ✓ Tire pressure is low
- ✓ Engine oil level is low
- ✓ Engine temperature is high

Vehicles have a **safety issue** and require **call** if one of the following conditions are true:

- ✓ Engine temperature: is high, outside temperature: is low
- ✓ Engine temperature: is low, outside temperature: is high

Based on the above requirements, we developed 2 separate models to identify anomalies,

1. Vehicle maintenance detection,
2. Vehicle recall detection.

In both these models, Principal Component Analysis (PCA) algorithm is used for anomaly detection.

5.3.1.1 Maintenance detection model

In the maintenance detection model, the model reports an anomaly if one of three indicators tire pressure, or engine temperature or engine oil, - satisfies its condition. As result, we only consider these three variables in building the model. In project Azure Machine Learning, we use a **Project Columns** module for selecting above three variables. Next we use the **PCA-based** anomaly detection module.

Principal Component Analysis (PCA) is technique in machine learning that can be applied to feature classification, selection, and anomaly identification. PCA converts case containing possibly correlated variables, into values called **principal components**. The idea behind using PCA-based modeling is to correlate data onto a lower-dimensional space(2-D) so that features and anomalies can be more easily identified.

In the case of anomaly detection, for every new input, the anomaly detector initial computes its projection on the eigenvectors, and so computes the normalized reconstruction error. This normalized error is that the anomaly score. In the maintenance detection downside, every record may be thought of as a degree during a third-dimensional house outlined by tire pressure, engine oil, and engine temperature coordinates. To capture these anomalies, we are able to project the first information within the third-dimensional onto a 2-dimensional house victimization PCA. Thus, we have a tendency to set the parameter variety of parts to use in PCA to be a pair of. This parameter plays a very important role in applying PCA-based anomaly detection. once protruding information victimization PCA, we are able to establish these anomalies additional simply.

Recall anomaly detection model. Anomaly detection model, we have a tendency to use the Project Columns and PCA-based anomaly detection modules during a similar method. Specifically, we have a tendency to initial extract 3 variables - engine temperature, outside temperature, and speed - victimization the Project Columns module. we have a tendency to

additionally embody the speed variable since the engine temperature generally is correlative to the speed. Next we have a tendency to use PCA-based anomaly detection module to project the information from the third-dimensional house onto a 2-dimensional house. The recall criteria square measure happy and then the vehicle needs recall once engine temperature and out of doors temperature square measure extremely negatively correlative. victimization PCA-based anomaly detection algorithmic rule, we are able to capture the anomalies once performing arts PCA.

Note that once coaching either model, we'd like to use traditional information that doesn't need maintenance or recall because the input file to coach the PCA-based anomaly detection model. within the rating experiment, we have a tendency to use the trained anomaly noticeion model to detect if the vehicle needs maintenance or recall.

5.3.2 Real-time analysis

The following SQL Query is used to get the average vehicle parameters like vehicle speed, engine temperature, fuel level, odometer reading, engine oil level, tire pressure, etc. to detect anomalies, alerts notification and determine the overall health conditions of vehicles operated in specific region and correlate it to demographics.

```
select BlobSource.Model, EventHubSource.city,count(vin) as cars, avg
(EventHubSource.engineTemperature) as engineTemperature, avg(EventHubSource.speed) as
Speed, avg(EventHubSource.fuel) as Fuel, avg(EventHubSource.engineoil) as EngineOil,avg
(EventHubSource.tirepressure) as TirePressure, avg(EventHubSource.odometer) as Odometer
into SQLSink from EventHubSource join BlobSource on EventHubSource.vin
= BlobSource.VIN group by BlobSource.model, EventHubSource.city, TumblingWindow(second,3)
```

Figure 23: Real-time processing: Stream analytics query

All the averages are calculated over a 3 seconds Window. TubmlingWindow as we require contiguous & non-overlapping time intervals.

5.3.3 Real-time prediction

An application is included as part of the solution to operationalize the machine learning model in real-time. This application called "RealTimeDashboardApp" is created and configured as part of the solution deployment. The application performs the following:

```
Select EventHubSource.vin, BlobSource.Model, EventHubSource.timestamp,EventHubSource.outsideTemperature,
EventHubSource.engineTemperature, EventHubSource.speed,EventHubSource.fuel, EventHubSource.engineoil,
EventHubSource.tirepressure, EventHubSource.odometer,EventHubSource.city,
EventHubSource.accelerator_pedal_position, EventHubSource.parking_brake_status,
EventHubSource.headlamp_status,
EventHubSource.brake_pedal_status,EventHubSource.transmission_gear_position,
EventHubSource.ignition_status,EventHubSource.windshield_wiper_status, EventHubSource.abs into
EventHubOut from EventHubSource join BlobSource on EventHubSource.vin = BlobSource.VIN
```

Figure 24: Stream analytics query for publishing the data to an output Event Hub instance

1. For every event that this application receives:

- Processes the data using Machine Learning Request-Response Scoring (RRS) endpoint. The RRS endpoint is automatically published as part of the deployment.
- The RRS output is published to a PowerBI dataset using the push APIs.

This pattern is also applicable in scenarios where you want to integrate a Line of Business application with the real-time analytics flow for scenarios such as alerts, notifications, messaging, etc.

5.4 Publish

5.4.1 Real-time analysis

One of the queries in the stream analytics job publishes the events to an output Event Hub instance.

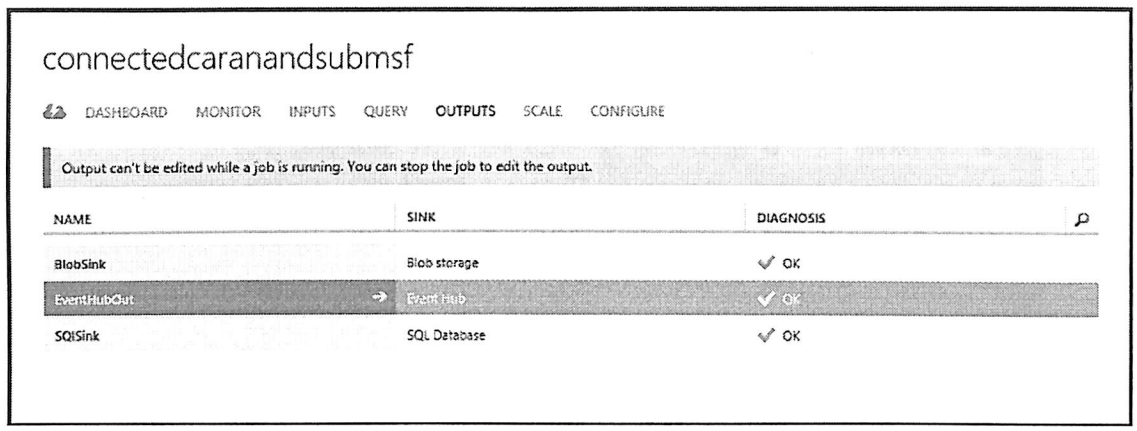


Figure 25: Stream analytics job publishes to an output Event Hub instance

```
Select EventHubSource.vin, BlobSource.Model, EventHubSource.timestamp,EventHubSource.outsideTemperature,
EventHubSource.engineTemperature, EventHubSource.speed,EventHubSource.fuel, EventHubSource.engineoil,
EventHubSource.tirepressure, EventHubSource.odometer,EventHubSource.city,
EventHubSource.accelerator_pedal_position, EventHubSource.parking_brake_status,
EventHubSource.headlamp_status,
EventHubSource.brake_pedal_status,EventHubSource.transmission_gear_position,
EventHubSource.ignition_status,EventHubSource.windshield_wiper_status, EventHubSource.abs into
EventHubOut from EventHubSource join BlobSource on EventHubSource.vin = BlobSource.VIN
```

Figure 26: Stream analytics query to publish to the output Event Hub instance

This stream of events is consumed by the RealTimeDashboardApp included in the solution. This application leverages the Machine Learning Request-Response web service for real-time scoring and publishes the resultant data to a PowerBI dataset for consumption.

5.4.2 Batch analysis

The results of the batch and real-time processing are published to the Azure SQL Database tables for consumption. The Azure SQL Server, Database and the tables are created automatically as part of the setup script.

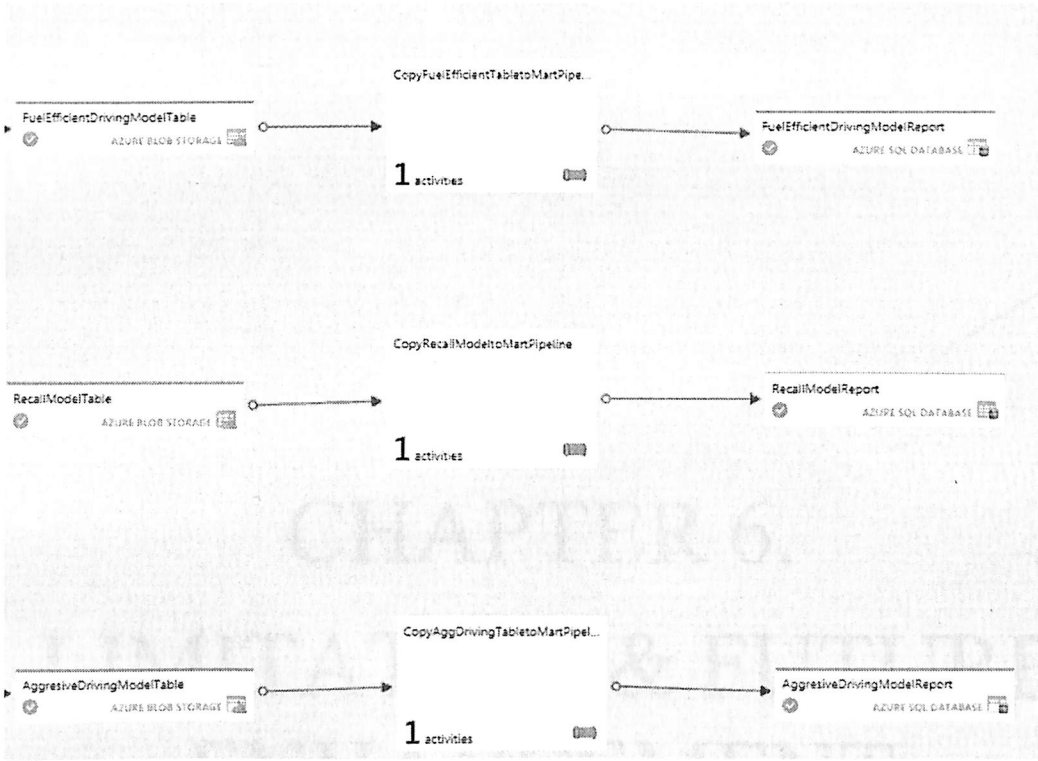


Figure 27: Batch processing results copy to data mart workflow

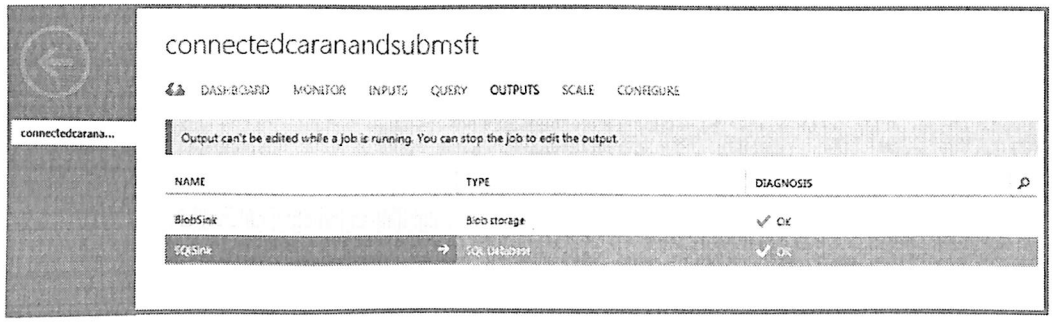


Figure 28: Stream analytics job publishes to data mart

CHAPTER 6.
LIMITATION & FUTURE ENHANCEMENT

6. Limitation & Future Enhancement

6.1 Limitation

6.1.1 Accidents Scenarios

The driver gap assist at stop-controlled intersections application area targets straight crossing path at non-signal and turn at non-signal multivehicle pre-crash scenarios, which include crashes at signalized intersections and driveways. This application area is targeted to the subset of crashes that occur at two-way stop-controlled intersections. Multi-vehicle crossing path crashes at two-way stop-controlled intersections can be classified as resulting from gap acceptance or stop sign violations. This application is intended for crashes resulting from poor gap acceptance and not from stop sign violations. The assumption is that the majority of cases in which the investigating officer coded the crash as a violation would involve a vehicle that failed to stop at the sign rather than a vehicle that stopped and then proceeded. Crashes resulting from stop sign violations (e.g., a driver is ticketed for violating a stop sign) were categorized in a separate pre-crash scenario, running stop sign.

6.1.2 Magnitude of Problem

There were an estimated 278,886 annual national target crashes based on weighted NASS GES data, and the estimated total annual cost of these crashes was nearly $18.3 billion. Additionally, 38 percent of the target crashes resulted in fatalities or injuries.

6.1.3 Relevant Distributions

Considering only the first two vehicles involved, the distribution of target crashes by the six vehicle-type categories indicated that crashes involving two light vehicles represented 93 percent of the total crashes. All other vehicle types represented only a small portion (2 percent or less each) of the involved vehicles. Crashes involving motorcycles accounted for 2 percent of multi-vehicle target crashes, which is slightly greater than the occurrence

of motorcycles in all multi-vehicle crashes. Motorcycles represented 1.4 percent of all multi-vehicle crashes.

The distribution of target crashes by area type indicated that the majority of the crashes (68 percent) occurred in urban areas. Roadways with five or more approach lanes represented 34 percent of the crashes. This is particularly noteworthy because it underscores the need for assistance with gap acceptance when crossing wider approaches. Roadways with two approach lanes represented 46 percent of the crashes, which can likely be attributed to the prevalence of this lane configuration.

6.1.4 Relationships in HSIS Data

HSIS data were investigated to determine the distribution of crash severity by intersection traffic control. This application area addresses potentially severe crashes at two-way stop-controlled intersections. The HSIS data support that these intersections represent a greater percentage of fatal and severe crashes when compared to other intersections. Of the 35,758 crashes at two-way stop-controlled intersections in California from 2005 through 2007, 1,543 crashes (4.4 percent) were fatal or severe injury. There were relatively fewer fatal and severe injury crashes at signalized and all-way stop-controlled intersections in terms of both frequency and percentage. Of the 32,925 crashes at signalized intersections, 639 crashes (1.9 percent) were fatal or severe injury. Of the 450 crashes at all-way stop-controlled intersections, 14 crashes (3.1 percent) were fatal or severe injury.

6.2 Future Work

Across industries and business disciplines, businesses use machine learning to increase revenue or reduce costs by performing tasks more efficiently than humans can do unaided. Included below are seven examples that demonstrate the versatility and wide applicability for machine learning.

6.2.1 Anti-theft. As you enter your car, a predictive model establishes your identity based on several biometric readings, rendering it virtually impossible for an imposter to start the engine.

6.2.2 Entertainment. Pandora plays new music it predicts you will like.

6.2.3 Traffic. Your navigator pipes up and suggests alternative routing due to predicted traffic delays. Because the new route has hills and your car's battery -- its only energy source -- is low, your maximum acceleration is decreased.

6.2.4 Breakfast. An en-route drive-through restaurant is suggested by a recommendation system that knows its daily food preference predictions must be accurate or you will disable it.

6.2.5 Social. Your Social Techretary offers to read you select Facebook feeds and Match.com responses it predicts will be of greatest interest. Inappropriate comments are accurately filtered out. CareerBuilder offers to read job postings to which you're predicted to apply. When playing your voicemail, solicitations such as robo call messages are screened by predictive models just like email spam.

6.2.6 Deals. You accept your smartphone's offer to read to you a text message from your wireless carrier. Apparently, they've predicted you're going to switch to a competitor, because they are offering a huge discount on the iPhone 13.

6.2.7 Internet search. As it's your colleague's kid's birthday, you query for a toy store that's en route. Siri, available through your car's audio, has been greatly improved -- better speech recognition and proficiently tailored interaction.

6.2.8 Driver inattention [8]. Your seat vibrates as internal sensors predict your attention has wavered -- perhaps you were distracted by a personalized billboard a bit too long.

6.2.9 Collision avoidance. A stronger vibration plus a warning sound alert you to a potential imminent collision -- possibly with a child running toward the curb or another car threatening to run a red light.

6.2.10 Reliability. Your car says to you, "Please take me in for service soon, as I have predicted my transmission will fail within the next three weeks."

Predictive analytics not only enhances your commute -- it was instrumental to making this drive possible in the first place:

6.2.11 Car loan. You could afford this car only because a bank correctly scored you as a low credit risk and approved your car loan.

Insurance. Sensors you volunteered to have installed in your car transmit driving behavior readings to your auto insurance company, which in turn plugs them into a predictive model in order to continually adjust your premium. Your participation in this program will reduce your payment by $30 this month.

6.2.12 Wireless reliability. The wireless carrier that serves to connect to your phone -- as well as your car -- has built out its robust infrastructure according to demand prediction.

6.2.13 Cyber-security. Unbeknownst to you, your car and phone avert crippling virus attacks by way of analytical detection.

6.2.14 Road safety. Impending hazards such as large potholes and bridge failures have been efficiently discovered and preempted by government systems that predictively target inspections.

6.2.15 No reckless drivers. Dangerous repeat moving violation offenders have been scored as such by a predictive model to help determine how long their licenses should be suspended.

6.2.16 Your health. Predictive models helped determine the medical treatments you have previously received, leaving you healthier today.

CHAPTER 7. CONCLUSION

7. Conclusion

Across industries, Stream Data technology has tremendous potential to leverage Machine Learning capabilities in enabling accurate decision-making for superior performance. There are many applications of Machine Learning techniques in the manufacturing industry, but successful implementation requires commitment from top management to enable changes in processes, active involvement of operational resources, availability of data, and collaboration with academia and technology partners with expertise in Machine Learning models and Stream Data technology. The solution for predictive maintenance analytics using Anomaly Detection & Principle Component Analysis, as presented in this document, demonstrates how Machine Learning can enable accurate prediction of failure events in the press line.

Recent developments in advanced computing, analytics, and low cost sensing have the potential to bring about a transformation in the manufacturing industry. The implementation of Machine Learning and Big Data may drive the next wave of innovation and may soon prove to be an unavoidable tactical move in achieving higher levels of optimization.

8. References

[1] L. Yu, D. J. Cleary and P. E. Cuddihy, "A Novel Approach to Aircraft Engine Anomaly Detection and Diagnostics," IEEE Aerospace Conference, Montana, USA, 2004.

[2] J.Schimert,"Data-DrivenFaultDetectionBasedon Process Monitoring and Dimension Reduction Techniques," IEEE Aerospace Conference, Montana, USA, 2008.

[3] T.Yairi,Y.Kawahara,R.Fujimaki,Y.SatoandK. Machida, "Telemetry-mining: A Machine Learning Approach to Anomaly Detection and Fault Diagnosis for Space Systems," IEEE Conference on Space Mission Challenges for Information Technology, 2006.

[4] A.T.Michael,S.Kumar,S.MathewandM.Pecht, "Anomaly Detection in Electronic Products," 2nd Electronics System Integration Technology (ESTC) Conference, Greenwich, UK, 2008.

[5] A.Saxena,B.WuandG.Vachtsevanos,"AHybrid Reasoning Architecture for Fleet Vehicle Maintenance," IEEE Instrumentation and Measurement Magazine, 2006.

[6] S. M. Namburu, M. Wilcutts, K. Choi, K. Pattipati, S. Chigusa and L. Qiao, "Systematic Data-driven Approach to Real-time Fault Detection and Diagnosis in Automotive Engines," IEEE Autotestcon, 2006.

[7] A.Routray,A.RajaguruandS.Singh,"DataReduction and Clustering Techniques for Fault Detection and Diagnosis in Automotives," IEEE Conference on Automation Science and Engineering (CASE), Canada, 2010.

Appendix A: Anomaly Detection in R

Time Series Forecasting in Azure ML using R

we are using Microsoft Azure Machine Learning Studio to build an experiment for doing time series forecasting using several classical time series forecasting algorithms available in R.

The main steps of the experiment are:

Step 1: Get data

Step 2: Split the data into train and test

Step 3: Run time series forecasting using R

Step 4: Generate accuracy metrics

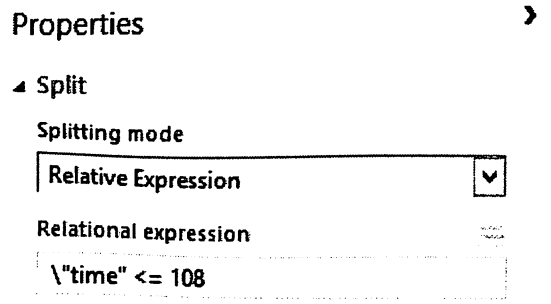
Step 5: Results

Step 1: Get data

We obtained the N1725 time series data from the publicly available M3 competition dataset, and uploaded the data to Azure ML Studio. This dataset has 126 rows and two columns, time and value.

Step 2: Split the data into train and test

We used the Split module in Azure ML Studio to divide the data into training and testing sets, using the Relational split option and specifying a time value as the split condition. We used the first 108 points for training and the remaining 18 points for testing the accuracy of various forecasting modules.



Step 3: Run time series forecasting using R

To compute forecasts, we used the following classical time series methods from the forecast package in R:

- Seasonal ARIMA
- Non Seasonal ARIMA
- Seasonal ETS
- Non -Seasonal ETS
- Average of Seasonal ETS and Seasonal ARIMA

For all seasonal methods, we used a seasonality value of 12.

The following R script was added to the Execute R Script module to build the model for seasonal ARIMA..

- Read the training data in dataset1 and the test data (for the timestamps) in dataset2.
- Create a ts object in R with the training data and specified seasonality.
- Learn a ARIMA model using the auto.arima() function from the forecast package in R.
- Compute the forecasting horizon by comparing the maximum timestamps in training and test datasets.

- Forecast using the learned ARIMA model for the computed horizon.

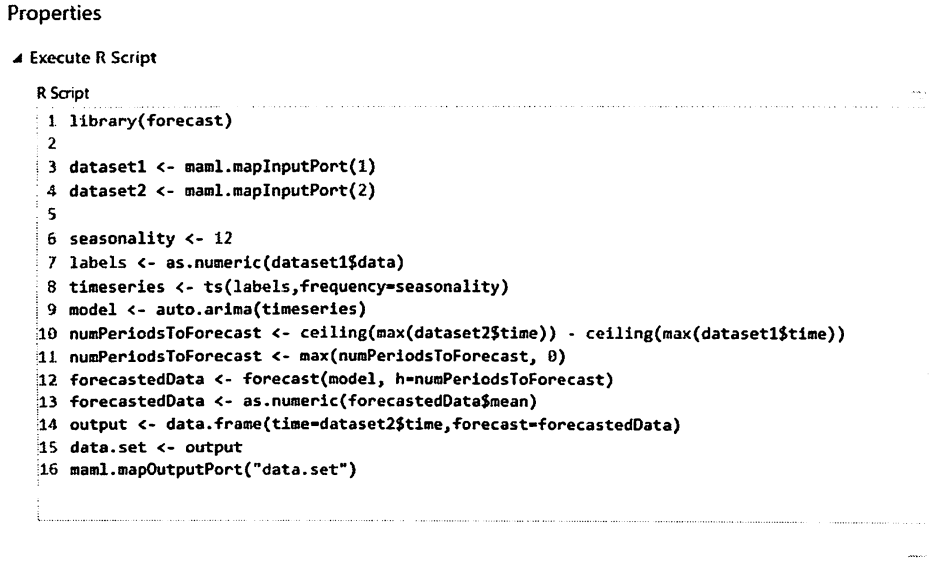


Figure 29: Sample R Code

For each of the other model types, we added a new Execute R Script module, added similar code to call the R packages with appropriate parameters.

Note: To save space, not all the scripts are shown here, but you can open the experiment in Azure ML Studio and click each module to see the R script details.

Step 4: Generate accuracy metrics

We joined the forecasting results from each of the methods with the test data, to compute the accuracy metrics. We used another instance of the Execute R Script module to compute the following metrics:

- Mean Error (ME) - Average forecasting error (an error is the difference between the predicted value and the actual value) on the test dataset
- Root Mean Squared Error (RMSE) - The square root of the average of squared errors of predictions made on the test dataset.
- Mean Absolute Error (MAE) - The average of absolute errors
- Mean Percentage Error (MPE) - The average of percentage errors

- Mean Absolute Percentage Error (MAPE) - The average of absolute percentage errors
- Mean Absolute Scaled Error (MASE)
- Symmetric Mean Absolute Percentage Error (sMAPE)

Step 5: Results

We found that the average of seasonal ETS and seasonal ARIMA models performs better than either of the two algorithms individually measured in terms of MASE/sMAPE/MAPE.

Method	ME	RMSE	MAE	MPE	MAPE	MASE	sMAPE
Seasonal ARIMA	-1.7	295.0	252.1574	-1.5	9.4	0.8	4.6
Non Seasonal ARIMA	-372.2	553.4	454.5	-15.8	18.2	1.5	8.0
Average Seasonal ARIMA & ETS	65.0	277.5	230.4	1.4	8.4	0.8	4.2
Seasonal ETS	131.7	322.5	255.7	4.3	9.3	0.8	4.9
Non Seasonal ETS	-344.7	533.6	438.5	-14.8	17.5	1.5	7.7

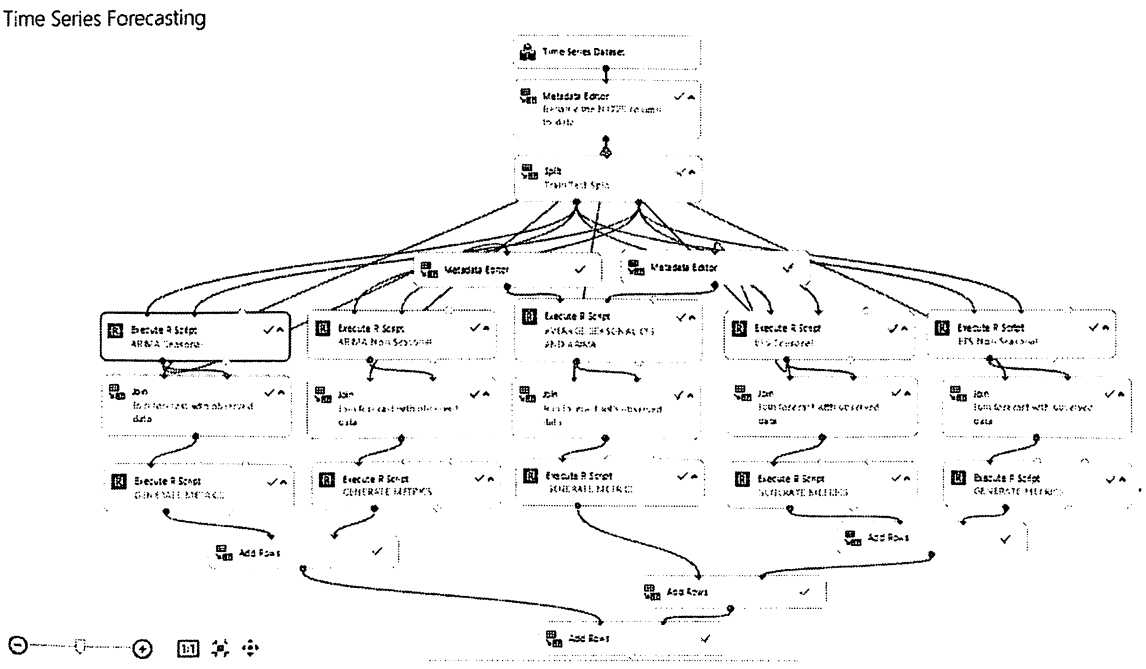


Figure 30: R Anomaly Detection

YV01

ORIGINALITY REPORT

9%	4%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	gallery.cortanaanalytics.com Internet Source	3%
2	Singh, Satnam, Clifton Pinion, and Halasya Siva Subramania. "Data-driven framework for detecting anomalies in field failure data", 2011 Aerospace Conference, 2011. Publication	2%
3	Domingos, Pedro. "A few useful things to know about machine learning", Communications of the ACM, 2012. Publication	1%
4	Submitted to Western Governors University Student Paper	<1%
5	Submitted to Infile Student Paper	<1%
6	www.lamboweb.com Internet Source	<1%
7	hadooptutorial.info Internet Source	<1%
8	anorien.warwick.ac.uk Internet Source	<1%

9	Azevedo, D. R., A. M. Ambrosio, and M. Vieira. "Applying Data Mining for Detecting Anomalies in Satellites", 2012 Ninth European Dependable Computing Conference, 2012. Publication	<1%
10	Submitted to Universiti Tenaga Nasional Student Paper	<1%
11	Submitted to City University of Hong Kong Student Paper	<1%
12	www.cloudcontactcenterzone.com Internet Source	<1%
13	Submitted to University of Technology, Sydney Student Paper	<1%
14	Submitted to University of Hertfordshire Student Paper	<1%
15	Submitted to Grand Valley State University Student Paper	<1%
16	Submitted to South Bank University Student Paper	<1%
17	www.monicawofford.com Internet Source	<1%
18	blog.djeepy1.net Internet Source	<1%