신뢰성응용연구
제18권, 제1호. pp.20-32, 2018

Anomaly Detection in Sensor Data

Jong-Min Kim[1] · **Jaiwook Baik**[2†]

[1]Statistics Discipline, Division of Science and Mathematics, University of Minnesota at Morris, U.S.A
[2]Department of Information Statistics, Korea National Open University, Seoul, Republic of Korea

Purpose: The purpose of this study is to set up an anomaly detection criteria for sensor data coming from a motorcycle.
Methods: Five sensor values for accelerator pedal, engine rpm, transmission rpm, gear and speed are obtained every 0.02 second from a motorcycle. Exploratory data analysis is used to find any pattern in the data. Traditional process control methods such as X control chart and time series models are fitted to find any anomaly behavior in the data. Finally unsupervised learning algorithm such as k-means clustering is used to find any anomaly spot in the sensor data.
Results: According to exploratory data analysis, the distribution of accelerator pedal sensor values is very much skewed to the left. The motorcycle seemed to have been driven in a city at speed less than 45 kilometers per hour. Traditional process control charts such as X control chart fail due to severe autocorrelation in each sensor data. However, ARIMA model found three abnormal points where they are beyond 2 sigma limits in the control chart. We applied a copula based Markov chain to perform statistical process control for correlated observations. Copula based Markov model found anomaly behavior in the similar places as ARIMA model. In an unsupervised learning algorithm, large sensor values get subdivided into two, three, and four disjoint regions. So extreme sensor values are the ones that need to be tracked down for any sign of anomaly behavior in the sensor values.
Conclusion: Exploratory data analysis is useful to find any pattern in the sensor data. Process control chart using ARIMA and Joe's copula based Markov model also give warnings near similar places in the data. Unsupervised learning algorithm shows us that the extreme sensor values are the ones that need to be tracked down for any sign of anomaly behavior.

Keywords: Sensor Data, Exploratory Data Analysis, Control Chart, ARIMA Model, Unsupervised Learning Algorithm

1. Introduction

Sensors are becoming popular in everyday life tasks as the era of Internet of Things (IoT) has lately arrived. The IoT will include 26 billion units installed by 2020 [28]. Future of IoT seems to expect each object in human life will be equipped with sensors which communicate each other to enable human life easier than before [6]. Sensors are deployed to monitor a phenomenon or to control a process [1]. They are used in diverse applications domains including business applications such as sales growth, industrial applications such as quality and reliability control of product, military applications such as enemy surveillance, and personal applications such as health monitoring [5].

Massive volume of IoT data generated by sensors is

† Corresponding Author jbaik@knou.ac.kr
Received October 12, 2017; revision received January 8, 2018; Accepted: January 9, 2018.

extremely dynamic, heterogeneous and imperfect [12]. It requires real-time analysis and decision making. One of the major goals of IoT systems is automatic monitoring and detection of abnormal events, changes or drift [13]. Traditionally anomaly detection had been carried out manually with the data visualization ([30]). But it is a burden on the operator. A survey is provided [13, 31, 34].

Wireless medical sensors collect various physiological parameter such as heart rate, pulse, oxygen saturation, Respiration and blood pressure. These sensors are attached to the subject's body and continuously monitored in hospital or home. Various sensor anomaly detection systems in medical sensors have been proposed and applied to date [37, 17, 20] .

Recently automated statistical and machine learning approaches such as minimum volume ellipsoid [35], convex pealing [35], nearest neighbor [33], clustering [8], neural network classifier [24], support vector machine classifier [9], and decision tree [23] have been employed. These methods are faster than manual approach but they are not suitable for real-time anomaly detection in streaming data. Real-time anomaly detection method has been employed in streaming environmental sensor data where incremental data-driven autoregressive model for the data was fitted and a prediction interval is calculated in order to identify streaming data anomalies [22].

A generic analytics engine is described [2] where sensor data that has been transmitted through cloud infrastructure is checked for stationarity and non-periodicity, and then nonparametric anomaly and change detection methods such as generalized Kolmogorov Smirnov test [18], bootstrapping based change detection [14] and one-class SVM are used to detect the abnormal behaviors.

Capturing all anomalies is impossible, which is the reason why anomaly detection methods are used in unsupervised setting [11]. But when there is a labelled response regression can be used to find relationship between a set of predictors and a response variable [3, 25]. There are other predictive models available. Kalman filter is a recursive filter that approximates the state of a system based on noisy measurements. Dynamic Baysian Networks can be seen as generalized Kalman filters or generalized hidden Markov models [22]. Artificial neural networks can be used to predict time series using historical data. Such networks can capture non-linear relationships and can be used for predicting financial time series [10].

In time series, a number of regression models such as Autoregressive (AR) [4], Autoregressive Moving Average (ARMA) [30], and Autoregressive Integrated Moving Average (ARIMA) [29] are applied to detect anomaly in the series. Sensor value from historic values is predicted using the spatiotemporal correlation that exists among physiological parameters and compared with the actual sensor value, where the difference is compared against a threshold value, which is dynamically adjusted [19]. However, models using external predictors such as ARX, ARMAS have not been thoroughly investigated in the literature.

Machine learning approaches are the Naïve Bayes, Bayesian network and decision tree methods [7]. Clustering method such as K nearest neighbor (K-NN) is also used. Mahalanobis Distance (MD) between predicted and actual multivariate instances is used to detect sensor anomaly [26]. With the arrival of a new instance MD is calculated between the training data in the sliding window and the current physiological parameter values. If MD is greater than the degree of freedom, abnormal physiological parameters are identified, and the window slides one slot by removing the oldest first instance and adding the new one.

Another sensor fault detection system for wireless sensor network is to utilize piecewise linear models of time series. Linear SVM is used to detect abnormal instances and linear regression is used for prediction purposes. Linear regression is a statistical modeling method used to predict the current value of the monitored parameters. One drawback to linear regression is that it is not an efficient prediction tool for application where the physiological parameters have rapid trend change.

Modeling serial dependence in time series is an important step in statistical process control. Fully auto-

mated routines for obtaining maximum likelihood estimates for given time series data and then drawing a Shewhart-type control chart have been proposed [27]. The routine is available as “Copula.Markov” package in R [15]. It has been pointed out that Joe's copula parametric maximum likelihood method provides the most reliable estimates of the UCL and LCL compared to the other copula methods. So we employed Joe’s copula to find any anomaly behavioral points in the sensor data.

Motorcycle is no exception to have a lot of sensor data. In this paper, we are looking for anomaly behavior in the motorcycle sensor data. Namely sensor values of accelerator pedal, engine rpm, transmission rpm, gear and speed are gathered every 20 millisecond for a total of about 39 minutes and examined for any anomaly behavior. In section 2, exploratory data analysis is tried for the sensor data in order to find any pattern or any correlations in the sensor data. In section 3, control charts for independent and correlated data are tried to find out-of-control points in the sensor data. Out-of-control points are interpreted as anomaly behavioral points in this section. Joe copula is tried to find any anomaly behavioral points. In section 4 unsupervised learning algorithm such as k-means is tried to find clusters among the sensor data. Lastly conclusions and some discussions are given in section 5. Readers must be warned that the results presented here had to be transformed in order to preserve the sensitive details of the data.

2. Exploratory Data Analysis

Every 20 millisecond sensor values of accelerator pedal, engine rpm, transmission rpm, gear and speed are obtained for a total of about 39 minutes. We want to find any anomalous behavior in the data. <Fig. 1> shows the time series plot of the raw data. In order to fine the trend in the data we can get an average of raw data for each of every second. <Fig. 2> gives the graph for secondly average accelerator pedal sensor values. As expected, <Fig. 2> gives more smooth representation of the accelerator pedal data. But since we want to find the anomalous behavior in the data we may as well use the original raw data in <Fig. 1>. So we will dwell on the original 20 millisecond sensor values in this paper.

Summary statistics are shown in <Table 1>. Accelerator pedal sensor has a mean of 9.783 while the median 0. The distribution of accelerator pedal sensor values is very much skewed to the left which resembles exponential distribution as shown in <Fig. 3>. Next, engine rpm sensor has similar mean and median values of about 1,160. According to <Fig. 1> and <Fig. 3>, most of the engine rpm sensor values are above 780 except at around time 40,000. It seems to us that the motorcycle

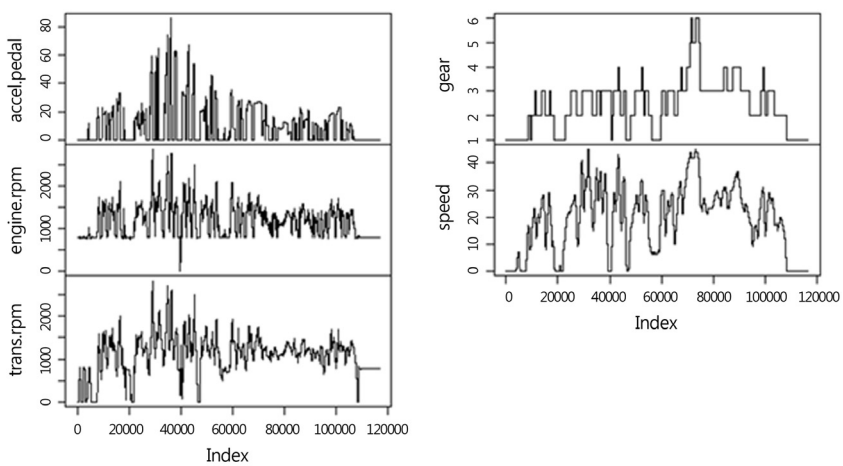

Fig. 1 Time series plot of the raw data

had been turned off during that time since both accelerator pedal and speed sensor values are 0. Next, transmission sensor has mean and median of 1086.2 and 1145.6. Next, gear sensor has mean and median values of 2.401 and 2. According to <Fig. 3>, the motorcycle had been at low gears of 1, 2 and 3 most of the time.

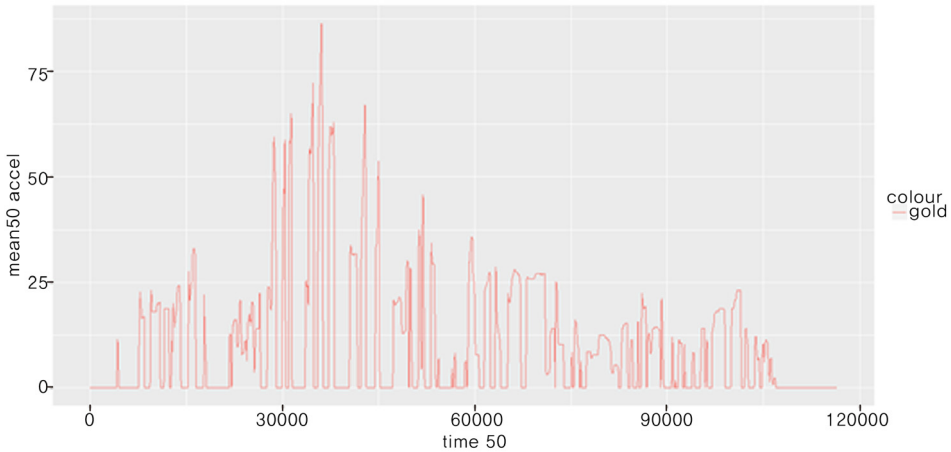

Fig. 2 Secondly average time series plot of accelerator pedal sensor data

Table 1 Summary statistics of the raw data

	accel.pedal	engine.rpm	trans.rpm	gear	speed
Min	0.000	0.0	0.0	1.000	0.00
1st Qu.	0.000	792.9	786.8	2.000	9.00
Median	0.000	1156.3	1145.6	2.000	22.00
Mean	9.783	1165.4	1086.2	2.401	19.26
3rd Qu.	16.351	1416.6	1334.8	3.000	28.00
Max.	86.275	2846.4	2812.9	6.000	45.00

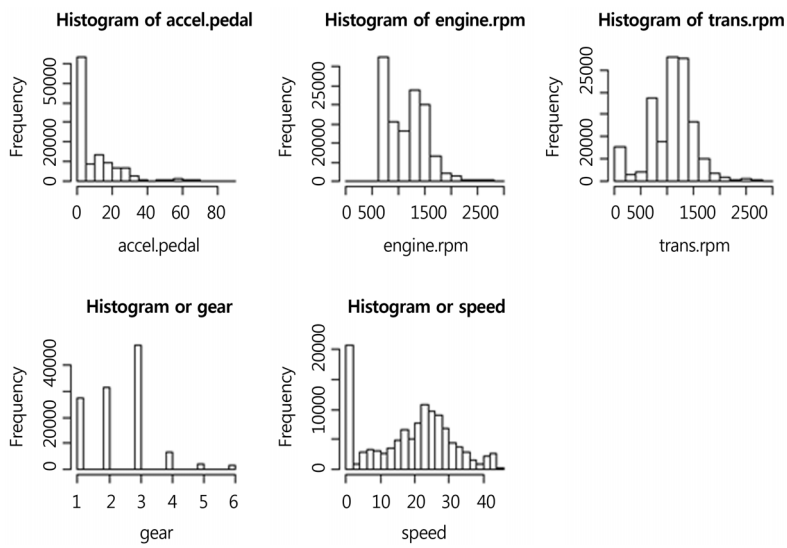

Fig. 3 Histogram of the 5 variables

Finally speed sensor has mean and median values of 19.26 and 22 respectively. <Fig. 3> shows that the motorcycle had been driven in city at speed less than 45 kilometers per hour.

It is interesting to see the correlations among the 5 variables since some of them are presumably correlated. <Table 2> gives the correlation between the variables while <Fig. 4> shows the graph of the correlation between the variables. For instance the correlation between gear and speed sensor values is as high as 0.85. It is obvious that the high value of gear means that the motorcycle is driving fast, which is reflected in the speed sensor gauge. Included in the high correlation values of 0.5 or above are between engine rpm and transmission rpm, between transmission rpm and speed, between accelerator pedal and engine rpm, and lastly between engine rpm and speed.

However, the correlations in <Table 2> are the correlations for each of the two variables measured at the same time. But as a driver if you drive a motorcycle you first start the engine, and then the engine sensor value goes up to 788 from 0 and the gear is at 1 even though accelerator, transmission rpm and speed sensor values are all 0. After several seconds of idling you press the accelerator pedal (and accelerator pedal sensor value goes up), and then in a split second the press of the accelerator pedal is transmitted to the engine (and the engine sensor value goes up), and then in another split second the power is transmitted to the transmission (and the transmission rpm value goes up), and then finally in another split second the motorcycle speeds up (and the speed sensor value goes up). So it may be interesting to know the correlations between one variable and another variable with time lags. But we will not pursue on this issue here.

Table 2 Correlation structure among 5 variables

	accel.pedal	engine.rpm	trans.rpm	gear	speed
accel.pedal	1.0000	0.6477	0.4638	0.0979	0.2676
engine.rpm	0.6477	1.0000	0.8195	0.1894	0.5582
trans.rpm	0.4638	0.8195	1.0000	0.4058	0.7342
gear	0.0979	0.1894	0.4058	1.0000	0.8537
speed	0.2676	0.5582	0.7342	0.8537	1.0000

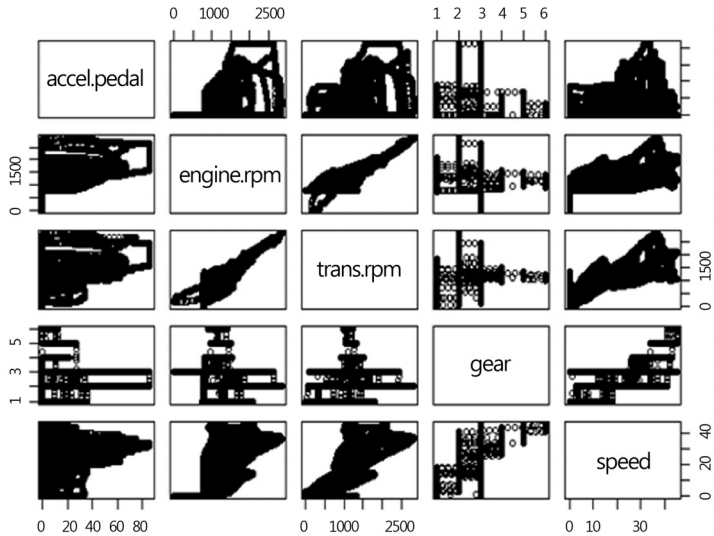

Fig. 4 Graph of the correlations between the two variables

3. Control Charts

We can treat anomaly detection as if we find any interesting points in a process. Control charts come in handy when controlling a process. So we'd like to plot the raw sensor data with center line and upper and lower control limits. Since it is not reasonable to form a homogeneous subgroup in this raw data we try individual X chart for each variable, for instance for accelerator pedal. Even though the raw data are highly correlated over time we treat them as if they are independent and draw X chart for accelerator pedal sensor values in <Fig. 5>. But since there is high correlation between adjacent sensor values the control limits based on the variability between adjacent sensor values does not fully reflect the process variability, thereby both lower control limit (9.709587) and upper control limit (9.857119) are very close together, which gives too many out-of-control warnings throughout almost all the period. We can see that there are 116116 out-of-control warnings from the total of 116593 points and that there are 115915 violating runs. The reason we have so many violating runs is because we have serially highly correlated sensor values.

It is very common that the serially observed sensor data are highly correlated, and some of them are not even stationary. If they are not stationary differenced series may yield the stationarity. Serially correlated data may be modelled through autoregressive and moving average (ARMA) model. Then the white noise which is the residual from the ARIMA model is used to produce X control chart to see where the abnormal pattern occurs. We apply the above procedure to accelerator pedal sensor values. It turned out that the appropriate model for the accelerator pedal sensor values is ARIMA (5, 1, 3). Hence, if we let be the original accelerator pedal sensor value at time and be the differenced sensor value, that is $Y_t = X_t - X_{t-1}$, then the appropriate ARIMA model turns out to be the following.

$$Y_t - 1.3343\,Y_{t-1} + 0.37\,Y_{t-2} + 0.1435\,Y_{t-3} - 0.2092\,Y_{t-4} + 0.0805\,Y_{t-5} = E_t - 0.1692E_{t-1} - 0.2176E_{t-2} + 0.226E_{t-3}$$

X control chart for the residual from the ARIMA (5, 1, 3) model is shown in <Fig. 6>. We can see that the upper and lower control limits are 0.032080967 and -0.0328096 respectively. It is obvious that the control limits are very close to center line since we have a very large number of observations, specifically 116593 to begin with. This time, however, there are only 6963 out-of-control warnings out of the total of 116593 points. This number is far less than 116116 which we had when we do not consider ARIMA model. But there are still too

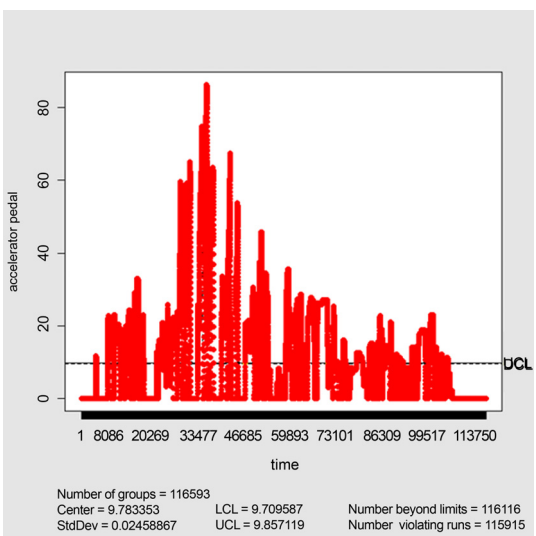

Fig. 5 X control chart for original accelerator pedal sensor data

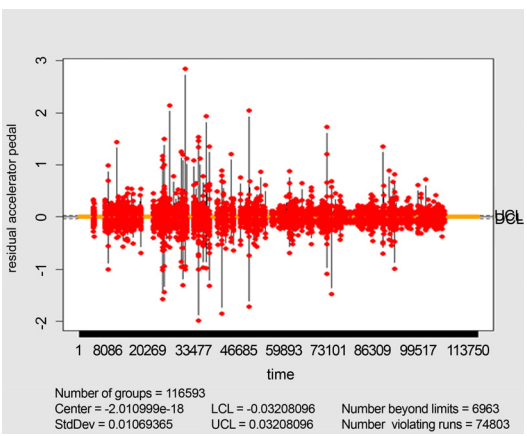

Fig. 6 X control chart of accelerator pedal for the residual from the ARIMA (5, 1, 3) model

many out-of-control warnings. So it would be helpful to see what abnormal events have happened for the cases where the residuals are above 2 in <Fig. 6>. Those cases were at times 26547, 30990 and 49704.

Copulas have been a popular method both for defining multivariate distributions and for modeling multivariate data [36] in the areas of actuarial science, bioinformatics, biostatistics and finance because a copula function does not require a normal distribution and independent, identical distribution assumptions. Furthermore, the invariance property of copula has been attractive in the finance area. A copula characterizes the dependence between the components of a multivariate distribution; they can be combined with any set of univariate marginal distributions to form a full joint distribution. A copula is a multivariate cumulative distribution function (CDF) whose univariate marginal distributions are all Uniform (0, 1). Suppose that $Y=(Y_1 \cdots, Y_d)$ has a multivariate CDF with continuous marginal univariate CDFs $F_{Y_1}(Y_1)$, $\cdots F_{Y_d}(Y_d)$. Then each of $F_{Y_1}(Y_1), \cdots F_{Y_d}(Y_d)$ is distributed according to Uniform (0, 1). Therefore the CDF of $\{F_{Y_1}(Y_1), \cdots F_{Y_d}(Y_d)\}$ is a copula. This CDF is called the copula of Y and denoted by C_Y. C_Y contains all information about dependencies among the components of Y but has no information about the marginal CDFs of Y. All d-dimensional copula functions C have domain $[0,1]^d$ and range $[0,1]$.

There are various types of copulas. One simple copula is independence copula. Multivariate normal and multivariate t-distributions offer a convenient way to generate families of copulas. In this paper we consider an Archimedean copula with the strict generator of the form below

$$C(u_1, \cdots u_d) = \phi^{-1}\{\phi(u_1 + \cdots + \phi(u_d)\}$$

where the generator function ϕ satisfies the following

1) ϕ is a continuous, strictly decreasing, and convex function mapping $[0,1]$ onto $[0, \infty]$
2) $\phi(0) = \infty$
3) $\phi(1) = 0$

There are several Archimedean copulas such as Frank copula, Gumbel copula and Joe copula. It's been pointed out that Joe's copula parametric maximum likelihood method provides the most reliable estimates of the UCL and LCL compared to the other copula methods. So we used Joe copula which has the generator $\phi_{Joe}(u|\theta) = -\log\{1-(1-u)^\theta\}$, $\theta \geq 1$. It turns out that θ is 2 for the accelerator pedal data. The estimates of the process mean and standard deviation are 9.783353 and 14.097014 respectively for our data. Therefore, the upper and lower control limits are $9.783353 + 3 \times 14.097014 = 52.074$ and $9.783353 - 3 \times 14.097014 = -32.508$. <Fig. 7> is the control chart after we fit Joe copula to our data. There are 3546 out-of-control warnings out of the total of 116593 points. This number is smaller than 6963 which we had when we considered ARIMA model. Out-of-control warning were at times 28549 to 28835, 30267 to 30420, 31033 to 36223, 37231 to 38077, 42480 to 42957, and 44899 to 45051. These out-of-control warning points do not exactly coincide with the warning points in ARIMA model but they are all close to each other.

In this section we tried univariate approach to accelerator pedal only. So it would be more appropriate to try multivariate approach to the data. We tried Hotelling's T^2 statistic to the data and came up with multivariate chart. But it was naïve to assume that we have independent observations. Multivariate time series analysis is desired where the concepts of cross correlation and transfer function models are used to characterize the original sensor data.

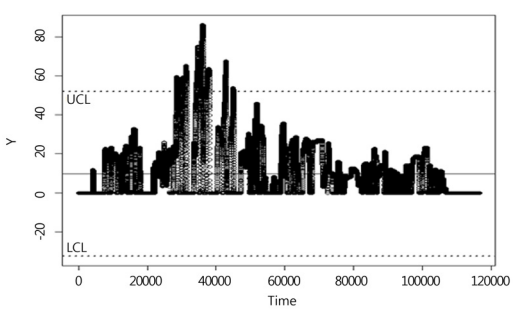

Fig. 7 Control chart by using Joe's copula to the accelerator pedal data

4. Unsupervised Learning

Unsupervised learning is a type of machine learning algorithm used to draw inferences from dataset consisting of input data without labeled responses. In our data, we have 5 sensor values, specifically sensor values of accelerator pedal, engine rpm, transmission rpm, gear and speed every 20 millisecond. We do not know whether the data at each time point is abnormal or not. So we can treat the data as if we are in an unsupervised learning environment. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modeled using a measure of similarity which is defined upon metrics such as Euclidean or probabilistic distance.

Common clustering algorithms are hierarchical clustering, k-means clustering, Gaussian mixture models, self-organizing maps, and hidden Markov models. Unsupervised learning methods are used in bioinformatics for sequence analysis and genetic clustering, in data mining for sequence and pattern mining, in medical image segmentation, and in computer vision for object recognition. In this paper, k-means clustering is tried for our data.

But before we try k-means clustering we try principal component analysis (PCA) to reduce our 5 dimensional data to a lower dimension to reveal the sometimes hidden, simplified dynamics that often underlie it. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables in a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the smaller of the number of original variables or the number of observations.

The result from the PCA is shown in <Table 3>. Hence, if we let x_{ij} be the j^{th} observation $(j=1,\cdots, 116593)$ of the i^{th} variable $(i=1, 2, 3, 4, 5)$ then the first principal component can be written as in the following equation, which resembles the overall mean or overall speed since high values of acceleration pedal, engine rpm, transmission rpm, gear and speed mean that the motorcycle is in overall high speed.

$$\begin{aligned} Z_{1j} = {} & 0.345(x_{1j-}\overline{x_1}) + 0.478(x_{2j-}\overline{x_2}) \\ & + 0.514(x_{3j} - \overline{x_3}) + 0.366(x_{4j} - \overline{x_4}) \\ & + 0.504(x_{5j} - \overline{x_5}) \end{aligned}$$

The second principal component can be written as in the following equation. This equation is the difference between the first 2 variables and the last two variables when ignoring the 3^{rd} variable since the coefficient of the 3^{rd} variable is small compared to the other coefficients. If we want to speed up we press the accelerator pedal and engine rpm goes up. Then in a split second gear goes up and the speed also goes up. So we can think of the first 2 variables as predecessor variables and the last 2 variables as successor variables. Then the second principal component can be thought as the difference between the predecessor and successor variables.

$$\begin{aligned} Z_{2j} = {} & 0.536(x_{1j} - \overline{x_1}) + 0.390(x_{2j} - \overline{x_2}) \\ & + 0.107(x_{3j} - \overline{x_3}) - 0.631(x_{4j} - \overline{x_4}) \\ & - 0.388(x_{5j} - \overline{x_5}) \end{aligned}$$

The above statements can be confirmed from the plot of the first two principal components as in <Fig. 8>. Note also in <Fig. 9> that the first two prominent principal components explain more than 80% of the total variance in the original data.

Table 3 The result from the PCA applied to our data

	PC1	PC2	PC3	PC4	PC5
accel.pedal	0.3451	0.5363	-0.7380	-0.1982	-0.0966
engine.rpm	0.4780	0.3899	0.2901	0.6934	0.2337
trans.rpm	0.5140	0.1075	0.4755	-0.6788	0.1932
gear	0.3664	-0.6313	-0.3803	0.0715	0.5634
speed	0.5039	-0.3877	0.0219	0.1185	-0.7624

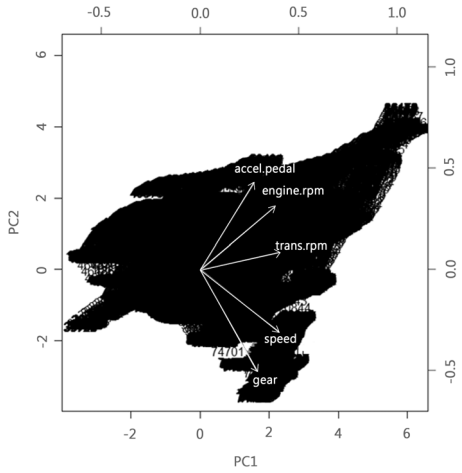

Fig. 8 First two principal components

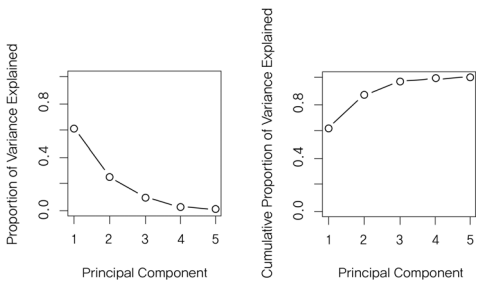

Fig. 9 Proportion of variance explained by principal components

K-means clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity. In k-means clustering, we have to specify the number k of clusters we want the whole data to be grouped into. The algorithm randomly assigns each observation to a cluster, and finds the centroid of each cluster. Then, the algorithm iterates through the following two steps:

Step 1: Reassign data points to the cluster whose centroid is closest.

Step 2: Calculate new centroid of each cluster.

K-means clustering with two clusters in terms of accelerator pedal and engine rpm is shown in <Fig. 10>. It seems to us that the cut-off value of 1000 in engine rpm divides the whole region into two clusters. In <Fig. 10>, accelerator pedal sensor values do not seem to do much in dividing the whole region into two clusters.

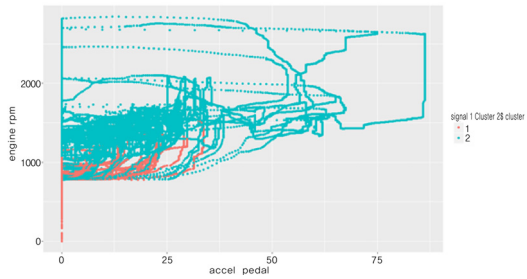

Fig. 10 Two clusters in terms of accelerator pedal and engine rpm

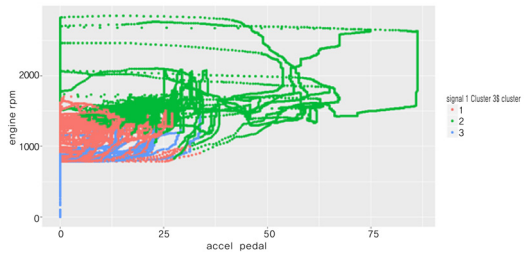

Fig. 11 Tree clusters in terms of accelerator pedal and engine rpm

K-means clustering with three clusters in terms of accelerator pedal and engine rpm is shown in <Fig. 11>. It seems to us that the second cluster in <Fig. 10> is subdivided into two disjoint clusters while the first cluster is still almost the same as before. The first cluster seems to be the region where the engine rpm is below 1000. The second cluster seems to be the region where accelerator pedal sensor value is below 25 and engine rpm sensor value is between 1000 and 1500. The rest of the whole region seems to belong to the third cluster. Four cluster are shown in <Fig. 12>. We can tell from <Fig. 11> and <Fig. 12> that the third cluster in <Fig. 11> seems to be again subdivided into two disjoint clusters, one cluster with accelerator pedal sensor values less than 30 and the other cluster with accelerator pedal sensor values greater than 30.

So far we have divided the whole region into two, three, and four disjoint regions. We found that large sensor values of accelerator pedal and engine rpm get subdivided into disjoint regions. So extreme sensor values are the ones that need to be tracked down for any sign of anomaly behavior in the sensor values. In the case of accelerator pedal and engine rpm, accelerator pedal sensor

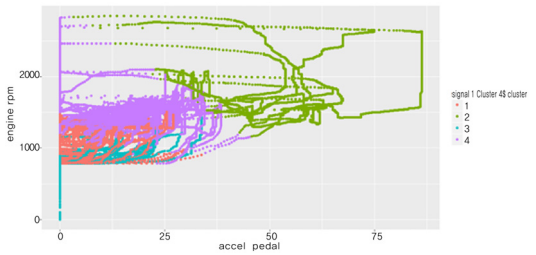

Fig. 12 Four clusters in terms of accelerator pedal and engine rpm

values greater than 35 and engine rpm sensor values greater than 1700 need to be watched for.

In <Fig. 10>～<Fig. 12> we divided the whole region into 2, 3 and 4 clusters in terms of accelerator pedal sensor values and engine rpm sensor values. We can do the same thing in terms of other sensor values. <Fig. 13>～<Fig. 15> show the same plots in terms of engine rpm and transmission rpm. According to <Fig. 13>, we can safely divide the whole region into two disjoint clusters; the first cluster with transmission rpm less than 900, the second cluster with transmission rpm greater than 900.

In <Fig. 14>, the second cluster in <Fig. 13> seems to be subdivided into two disjoint clusters; one cluster with engine rpm less than 1400 and transmission rpm less than 1500, the other cluster with the rest of the region. This 'rest of the region' is subdivided again into two disjoint clusters in <Fig. 15>, making four disjoint clusters all together.

We found again that large sensor values of engine rpm and transmission rpm get subdivided into two, three, and four disjoint regions. So extreme sensor values are the ones that need to be tracked down for any sign of anomaly behavior in the sensor values.

Principal component analysis is a method of extracting important variables from a large set of variables available in a data set. It extracts low dimensional set of features from a high dimensional data set with a motive to capture as much information as possible as shown in <Fig. 9>. Now we could draw a graph of two clusters in terms of the first two principal component scores z_{1j} and z_{2j} as in <Fig. 16>. The two clusters are roughly divided by the cut-off line of the first principal component score of -0.7. So the first cluster is the left hand side of $z_{1j} = -0.7$ while the second cluster is the right hand side of $z_{1j} = -0.7$.

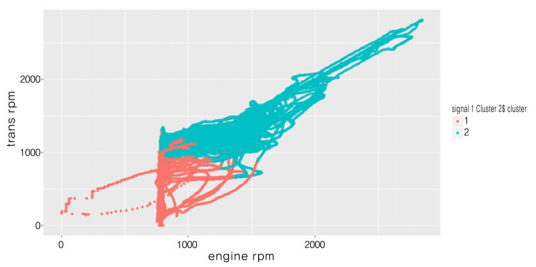

Fig. 13 Two clusters in terms of engine rpm and transmission rpm

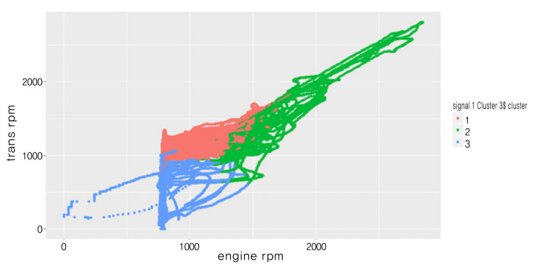

Fig. 14 Three clusters in terms of engine rpm and transmission rpm

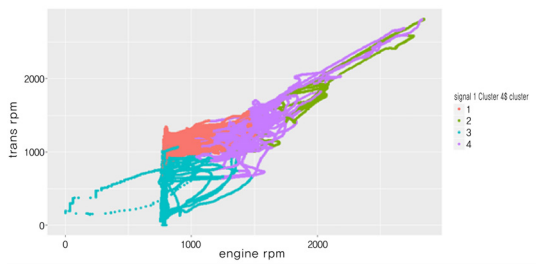

Fig. 15 Four clusters in terms of engine rpm and transmission rpm

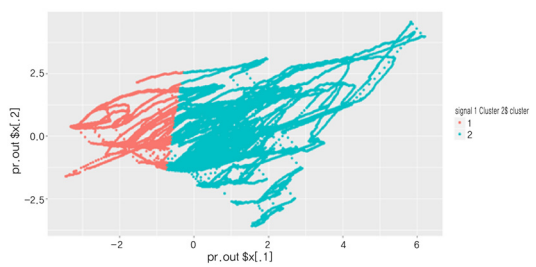

Fig. 16 Four clusters in terms of engine rpm and transmission rpm

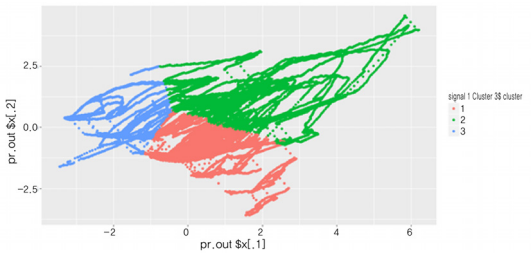

Fig. 17 Three clusters in terms of principal components 1 and 2

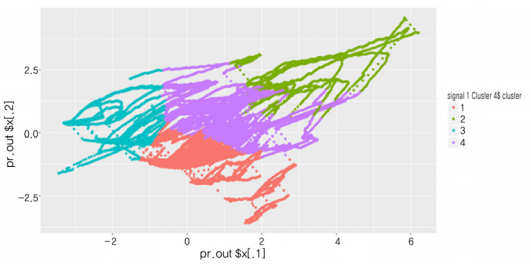

Fig. 18 Four clusters in terms of principal components 1 and 2

But if we want to have 3 disjoint clusters then the second cluster in <Fig. 16> seems to be subdivided into two disjoint clusters, namely one cluster with the second principal component score less than 0 and the other cluster with the second principal component score greater than 0. Finally if we want to have 4 disjoint clusters then the top right hand side in <Fig. 17> seems to be subdivided into two disjoint clusters as in <Fig. 18>.

We found that large first and second principal component scores get subdivided into two, three and four disjoint regions. So extreme principal component scores are the ones that need to be tracked down for any sign of anomaly behavior. In this case first principal component score greater than 2 and second principal component score greater than 1 need to be watched for.

5. Conclusion and Comments

Every 20 millisecond sensor values of accelerator pedal, engine rpm, transmission rpm, gear and speed are obtained for a total of about 39 minutes. Exploratory data analysis is used to find any pattern in the data. For instance, the distribution of accelerator pedal sensor values is very much skewed to the left. The motorcycle seemed to have been driven in city at speed less than 45 kilometers per hour. Included in the high correlation values are between gear and speed, between engine rpm and transmission rpm, between transmission rpm and speed, between accelerator pedal and engine rpm, and between engine rpm and speed.

Next, traditional process control charts such as X control chart fails due to severe autocorrelation in each sensor data. ARIMA model has been fitted and X control chart to the residuals from the fitted ARIMA model are used to find any anomaly behavior in the sensor data. In the case of accelerator pedal, there are three points where they are beyond 2 sigma limits. So it would be useful if we can find why they are out of control. We can also use copula based Markov model for each sensor value in order to find any anomaly behavior in the data. We found that the two approaches give approximately close points as out of control points.

Finally unsupervised learning algorithm such as k-means clustering is used to find any anomaly spot in the sensor data. K-means clustering has been done in terms of accelerator pedal and engine rpm, and in terms of engine rpm and transmission rpm. We found that large sensor values get subdivided into two, three, and four disjoint regions. So extreme sensor values are the ones that need to be tracked down for any sign of anomaly behavior in the sensor values.

In this paper the methods we have used require a lot of time to process the data since we are dealing with the whole data set. In fact, carrying and analyzing the whole data set at all times would be a hindrance to real-time analysis and decision making. One way to get over this problem is naturally break the whole data set into moving windows, thereby dealing with, say minutely data so that we can apply process control techniques such as control chart and clustering algorithms k-means and hierarchical clustering to the minutely data.

We could treat speed as a labelled response since if we want a high speed, then we will press accelerator pedal. And then in a successive split second engine,

transmission, and gear sensors will go up to make the speed go up finally. So if we let y_t be the speed and $x_{1,t}$, $x_{2,t}$, $x_{3,t}$, $x_{4,t}$ be the accelerator pedal, engine rpm, transmission rpm, and gear at time t then we can construct the regression model as follows:

$$\begin{aligned} y_t = & a_{1,t}x_{1,t} + a_{1,t-1}x_{1,t-1} + \cdots \\ & + a_{2,t}x_{2,t} + a_{2,t-1}x_{2,t-1} + \cdots \\ & + a_{3,t}x_{3,t} + a_{3,t-1}x_{3,t-1} + \cdots \\ & + a_{4,t}x_{4,t} + a_{4,t-1}x_{4,t-1} + \cdots \end{aligned}$$

Then for every minutely moving window we can determine the above regression model and decide whether there is any outlier in y. We can treat those outliers as anomaly behavior in the data. We can also find large leverage points in x and treat them as anomaly behavioral points as well. Those are a few future research areas.

References

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). "Wireless sensor networks: A survey". Computer Networks, Vol. 38, pp. 393-422.

[2] Amitai, A., Gilad, W., and Lev, F. (2016). "Change and anomaly detection framework for Internet of Things data streams". Intel, https://software.intel.com/en-us/articles/change-and-anomaly-detection-framework-for-internet-of-things-data-streams.

[3] Anderson, T. W. (2011). "The statistical analysis of time series". John Wiley & Sons, Vol. 19.

[4] Anscombe, F. J. (1960). "Rejection of outliers". Technometrics, Vol. 2, No. 2, pp. 123-146.

[5] Arampatzis, T., Lygeros, J., and Manesis, S. (2005). "A survey of applications of wireless sensors and wireless sensor networks". In Proceedings of the 2005 IEEE international symposium on intelligent control and 13th Mediterranean conference on control and automation, Limassol, Cyprus, 27-29 June 2005, pp. 719-724.

[6] Atzori, L., Iera, A., and Morabito, G. (2010). "The Internet of Things: A survey". Computer Networks, Vol. 54, pp. 2787-2805.

[7] Bishop, C. M. (2006). "Pattern recognition and machine learning". Springer: New York, NY.

[8] Bolton, R. J. and Hand, D. J. (2001). "Unsupervised profiling methods for fraud detection". In Proceedings of Credit Scoring and Credit Control VII, Edinburgh, UK, 5-7.

[9] Bulut, A., Singh, A. K., Shin, P., Fountain, T., Jasso, H., Yan, L., and Elgamal, A. (2005). "Real-time nondestructive structural health monitoring using support vector machines and wavelets". In Meyendorf, N., Baakline, G.Y., Michel, B. (Eds.), Proceedings of the SPIE 5770, Advanced Sensor Technologies for Nondestructive Evaluation and Structural Health Monitoring, pp. 180-189.

[10] Chakraborty, K., Mehrotra, K., Mohan, C. K., and Ranka, S. (1992). "Forecasting the behavior of multivariate time series using neural networks". Neural networks, Vol. 5, No. 6, pp. 961-970.

[11] Chandola, V., Banerjee, A., and Kumar, V. (2009). "Anomaly detection: A survey". ACM Computing Surveys (CSUR), Vol. 41, No. 3, pp. 1-58.

[12] Chen, Y. K. (2012). "Challenges and opportunities of internet of things", 17th Asia and South Pacific Design Automation conference, pp. 383-388. http://doi.org/10.1109/ASP-DAC.2012.6164978.

[13] Chui, M., Loffler, M., and Roberts, R. (2010). "The Internet of Things". Mckinsey Quarterly.

[14] Dasu, T., Krishnan, S., Venkatasubramanian, S., and Yi, K. (2006). "An information-theoretic approach to detecting changes in multi-dimensional data stream". In Proceedings of the Symposium on the Interface of Statistics, Computing Science and Applications.

[15] Emura, T., Long, T. H., and Sun, L. H. (2017). "R routines for performing estimation and statistical process control under copula-based time series models". Communications in Statistics-Simulation and Computation, Vol. 46, pp. 3067-3087.

[16] Fox, A. J. (1972). "Outliers in time series". Journal of the Royal Statistical Society. Series B (Methodological), pp. 350-363.

[17] Fritsch, V., Varoquaux, G., Thyreau, B., Poline, J. B.,

and Thirion, B. (2012). “Detecting outliers in high-dimensional neuroimaging datasets with robust covariance estimators”. Medical Image Analysis, Vol. 16, No. 7, pp. 59-1370.

[18] Glazer, A., Lindenbaum, M., and Markovitch, S. (2012). “Learning high-density regions for a generalized Komogorov-Smirnov Test in high-dimensional data”. In Advances in neural information processing systems 25 (NIPS 2012).

[19] Haque, S. A., Rahman, M., and Aziz, S. M. (2015). “Sensor anomaly detection in wireless sensor networks for healthcare”. Sensors, Vol. 15, pp. 8764-8786.

[20] Hauskrecht, M., Batal, I., Valko, M., Visweswaran, S., Cooper, G. F., and Clermont, G. (2013). “Outlier detection for patient monitoring and alerting”. Journal of Biomedical Informatics, Vol. 46, pp. 47-55.

[21] Hill, D. J., Barbara, S., and Minsker, S. (2010). “Anomaly detection in streaming environmental sensor data: A data-driven modeling approach”. Environmental modelling and software, Vol. 25, No. 9, pp. 1014-1022.

[22] Hill, D. J., Minsker, B. S., and Amir, E. (2007). “Real-time Bayesian anomaly detection for environmental sensor data”. In Proceedings of the Congress-International Association for Hydraulic Research, Citeseer, Vol. 32.

[23] John, G. H. (1995). “Robust decision trees: removing outliers from databases”. In Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, pp. 174-179.

[24] Kozuma, R., Kitamura, M., Sakuma, M., and Yokoyama, Y. (1994). “Anomaly detection by neural network models and statistical time series analysis”. In Neural Networks 1994. IEEE World Congress on Computer Intelligence, Orlando, FL.

[25] Lipsey, M. W. and Wilson, D. B. (2001). “Practical meta-analysis”. Sage publications Thousand Oaks, CA, Vol. 49.

[26] Liu, F., Cheng, X., and Chen, D. (2007). “Insider Attacker Detection in Wireless Sensor Networks”. In Proceedings of 26th IEEE International Conference on Computer Communications, Anchorage, AK, USA, 6-12 May 2007, pp. 1937-1945.

[27] Long, T. H. and Emura, T. (2014). “A control chart using copula based Markov chain models”. Journal of the Chinese Statistical Association, Vol. 52, pp. 466-496.

[28] Middleton, P., Kjeldsen, P., and Tully, H. (2013). “Forecast: the Internet of Things”. Worldwide, Gartner.

[29] Moayedi, H. Z. and Masnadi-Shirazi, M. A. (2008). “ARIMA model for network traffic prediction and anomaly detection”. In 2008 International Symposium onInformation Technology, Vol. 4, pp. 1-6.

[30] Mourad, M. and Bertrand-Krajewski, J. (2002). “A method for automatic validation of long time series of data in urban hydrology”. Water Science and Technology, Vol. 45, pp. 263-270.

[31] Patcha, A. and Park, J. (2007). “An overview of anomaly detection techniques: Existing solutions and latest technological trends”. Computer Networks, Vol. 51, pp. 3448-3470.

[32] Pincombe, B. (2005). “Anomaly detection in time series of graphs using ARMA processes”. ASOR BULLETIN, Vol. 24, No. 4, pp. 2-10.

[33] Ramaswamy, S., Rastogi, R., and Shim, K. (2000). “Efficient algorithms of mining outliers from large data sets”. In Proceedings of the ACM SIGMOD Conference on Management of Data, Dallas TX, pp. 427-438.

[34] Rassam, M. A., Zainal, A., and Maarof, M. A. (2013). “Advancements of data anomaly detection research in wireless sensor networks: A survey and open Issues”. Sensors, Vol. 13, pp. 10087-10122.

[35] Rousseeuw, P. J. and Leroy, A. M. (2003). “Robust regression and outlier detection“. John Wiley & Sons, New York.

[36] Rupert, D. and Matteson, D. S. (2015). “Statistics and data analysis for financial engineering with R examples”. Springer.

[37] Salem, O., Guerassimov, A., Mehaoua, A., Marcus, A., and Furht, B. (2013). “Sensor fault and patient anomaly detection and classification in medical wireless sensor networks”. In Proceedings of 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, 9-13 June 2013, pp. 4373-4378.