Layer: An Alternative Approach To Solve Large Capacitated Vehicle Routing Problem with Time Window Using AI and Exact Method

Mukherjee, Krishnendu

Associate Professor, Woxsen University

3 June 2023

Layer: An Alternative Approach To Solve Large Capacitated Vehicle Routing Problem with Time Window Using AI and Exact Method

Abstract: To the best of my knowledge, this problem has never been addressed by any researcher. This paper studies the effect of K-means, the Gaussian Mixture Model (GMM), and the integrated use of autoencoder and K-means on the computational time, MIP gap, feasible route, subtour, and the optimum use of vehicles. Miller-Tucker-Zemlin (MTZ) subtour elimination constraint is considered in this regard. This paper also gives the concept of a "layer", which could be effective to solve a large vehicle routing problem with a time window (VRPTW) quickly.

Keyword: Machine Learning, Deep Learning, Mixed Integer Linear Program, and Large VRPTW

1. Introduction

Machine learning (ML), business analysis (BA), optimization, artificial intelligence etc are commonly used to implement Industry 4.0 for maximizing automation (Maryam and Krishnan, 2020). Thus, prescriptive analysis is one of the key features of Industry 4.0. Industry 4.0 guarantees the maximum uptime throughout the production chain and enhances productivity by decreasing the production cost (Sahal *et* al., 2020). Industry 4.0 is the application of big data analytics for real-time monitoring (Bellavista *et* al., 2019, Peres *et* al.,2017). Precisely, real-time streaming data analytics is one of the central themes of Industry 4.0. Digitalization is the backbone of Industry 4.0 and Smart Manufacturing. Smart Manufacturing and Industry 4.0 are often used as a misnomer. They are basically related concepts. Smart Manufacturing is a subset of Industry 4.0. Both Industry 4.0 and Smart Manufacturing needs an effective real-time locating/tracking system (RTLS) for monitoring inbound, outbound, and in-transit logistics to give optimum material flow to the end-user. RTLS can be used to monitor harsh braking, over speeding, idle time, speed, ignition status of a particular vehicle along with its latitude and longitude in real-time. A quantum leap in the logistics sector has been observed due to the extensive use of the global positioning system (GPS) and advanced map matching algorithms, advancement of computer processors (such as CUDA, Quantum Processor etc), effective use of big data analytics and the cloud etc. Recently researchers have considered vehicle routing problem with time window (VRPTW), a combinatorial optimization problem (COP), to use advanced methods such as attention-based models (Welling *et* al.,2019), reinforcement learning (Nazari *et* al.,2018), graph embedding model (Song *et* al.,2016), adaptive large neighborhood search (ALNS)(Ropke and Pisinger,2006),hyper-heuristic method (Bai *et* al.,2007), neural large neighborhood search (NLNS) (Hottung and Tierney,2019) etc. Vehicle route optimization can reduce logistics lead time, waiting time of the vehicle, idle time of the driver, transportation costs etc. Customer satisfaction, on the other hand, is an integrated part of the route optimization problem (Ghannadpour and Hooshfar, 2015, Barkaoui *et* al.,2015, and Fang,2011). Clustering based method has extensively been used for capacitated vehicle routing problem (CVRPTW) and vehicle routing problem with time window (VRPTW) (Battarra *et* al., 2002, Barreto *et* al.,2007, Alesiani *et* al., 2022, Abbas and Fan, 2018, Song *et* al., 2013). The literature review shows that the majority of the clustering based approaches are integrated with metaheuristics such as ant colony optimization (ACO), genetic algorithm (GA), iterated local search (ILS), memetic algorithm etc. Only the work of Battarra *et* al. (2002) is on the exact method but they used their method for CVRP.

This study attempts to answer the following four main research questions:

RQ1. Can the exact method solve a large vehicle routing problem with time window (VRPTW)? What would be the computational time?

RQ2. Can the exact method be integrated with clustering methods to give near-optimal or optimal solutions?

RQ3. What would be the effect of the number of clusters on the MIP gap, computational time, subtour, feasible route, and the required number of vehicles?

RQ4. Can the proposed method be used with a real-time locating/tracking system (RTLS), which is essential for Industry 4.0?

These four main research questions determine the performance of the route optimization software in the context of Industry 4.0: first, the proposed approach can be digitalized to give better insight to the end-user. The proposed approach can be used for real-time data. Second, it highlights key features to give a feasible and optimum route, and, third, it gives an alternative economical way to digitalize vehicle route optimizer for Smart Manufacturing and Industry 4.0 .

The literature review didn’t find any research paper to answer the above questions. A Gehring and Homberger's 1000 customer instance is considered in this regard to study the effect of clusters obtained from K-means, GMM, and autoencoder based K-means. Initially, a two-indexed BigM model has been developed and tested for the ‘R’ and ‘C’ series of Solomon’s VRPTW instances and the same baseline model has been used for a 1000 customer instance. This paper consists of four main sections. Section 1 A brief introduction to VRPTW has been given in the context of Industry 4.0. Section 2 Two-index BigM model for VRPTW has been formulated as a baseline mathematical model. This section also discusses in brief K-means, GMM, and autoencoder based K-means. Section 3 A detailed discussion has been given to analyze the results to give answers to the above research questions. Section 4 gives a conclusion.

2. Two-index BigM model for VRPTW

The mixed Integer Linear Program (MILP) and BigM formulation, the exact method, are quite popular in the industry for solving VRPTW. However, metaheuristic is more popular than the exact method (De Jaegere *et* al.,2014). The expected reasons could be the high computational time of the exact method, difficulties in building the mathematical model, problem size, interest of researchers to explore new methods in the route optimization domain, application areas etc. For ex., Metaheuristic is quite a popular method for mobile-based route optimization software (a.k.a App).

The proposed two-index BigM model for VRPTW as follows:

Let G = (V, A) be a diagraph where V = $\{0\} \cup N \; and \; N = \{1,2,3,4,5, \ldots n\}$ and n is the number of customers and A = $\{(i,j) | \; i,j \; \in V, i \neq j \}$ and '0' refers to depot. A is the arc and it connects two customers. Each arc (i,j) is associated with a non-negative cost $\left(c_{ij}\right)$. It is basically a Euclidean distance. It is pertinent to mention that this model needs further modification to be used for GPS data. Raw GPS data give inaccurate position due to the presence of tall buildings, flyover, canyon etc. Raw GPS data can be processed using Hidden Markov Model (HMM), known as Map Matching, to get the exact route and the distance traversed by the vehicle, shown in fig 1. Interested readers can refer relevant research papers in this regard.

Fig 1 (a) Raw GPS data (b) Processed GPS data after using HMM (c) Original point i.e. raw GPS data and processed data with projection using available python package.

Assumptions:

Let us consider the set of homogenous vehicles. Each vehicle has the same maximum capacity. The average speed of the vehicle is 60 km/hour. Vehicles will start the trip from a predefined depot and return to the depot after completing the trip. The objective is to reduce the number of vehicles, total distance traversed, and waiting time.

Notations:

n: number of customers
N: Set of customers = {1,2,3,…,n}
V: Set of vertices or nodes = $\{0\} \cup N$
A: Set of Arcs = $\{(i,j) \in V^2 : i \neq j\}$
$C_{i,j}$ $is\ the\ distance\ or\ length\ of\ arc(i,j) \in A$
$CCap$: $Maximum\ weight\ that\ can\ be\ carried\ by\ the\ vehicle$
q_i : $Amount\ that\ can\ be\ deivered\ to\ the\ customer\ i\ \in N$
$RTime_i$: $Ready\ Time\ of\ the\ customer\ \ i\ \in N$
$STime_i$: $Service\ time\ to\ the\ customer\ i\ \in N$
$DDate_i$: $Due\ date\ of\ the\ customer\ i\ \in N$
w_i: $Waiting\ time\ of\ the\ vehicle\ at\ the\ customer\ i\ \in N$
$LENV$: $Length\ of\ vertex$
$X_{i,j}$: $1\ if\ vehicle\ travels\ along\ the\ arc\ (i,j) \in A\ \ else\ 0$
BigM=999999

Objective function:

$$Minimize\ \sum_{i=1}^{n} \sum_{j=1, i\neq j}^{n} C_{i,j} X_{i,j}\ + 10 \sum_{i=1}^{n} w_i + \sum_{j=1}^{n} X_{0,j}$$

s.t.

$\sum_{j \in V, j\neq i} X_{i,j} = 1\ \forall\ i \in N$ …………………(1)
$\sum_{i \in V, i\neq j} X_{i,j} = 1\ \forall\ j \in N$ ……………………(2)
$u_i + q_j + BigM * X_{i,j}\ \ \leq u_j + BigM \quad \forall\ i,j\ \in A\ , i \neq j\ and\ j \neq 0$ …………(3)
$t_i + STime_i + C_{i,j} + BigM * X_{i,j}\ \ + w_j \leq t_j + BigM \quad \forall\ i,j\ \in A\ , i \neq j\ and\ j \neq 0$…….(4)
$U_i - U_j + (LENV - 1) * X_{i,j}\ \leq (LENV - 2)$ …………………(5)

$X_{i,j}\ \in \{0,1\}\ \forall\ i,j\ \in A$

Bounds:

$\boldsymbol{q_i \leq u_i \leq CCap}$
$RTime_i \leq \boldsymbol{t_i} \leq DDate_i$

In the above objective function, the first term refers to the shortest distance to traverse and the second term refers to the minimization of the waiting time of the vehicle at each node and the last term refers to the use of the minimum number of vehicles. Eq. 1 means that a vehicle will travel from i^{th} to any of the j^{th} nodes, but not all. Similarly, vehicles will travel from j^{th} node to any of the i[th] nodes, but not all. Node will be selected based on the shortest distance. Eq.1 and Eq. 2 basically confirm that a vehicle will visit each customer only once. Eq. 3 refers to the cumulative demand that must be less than the maximum capacity of a vehicle. Eq. 4 refers to the arrival time of any vehicle that should be in between the predefined time window. Eq.5 is popularly known as *MTZ Subtour Elimination Constraint.* The above mathematical model can solve up to 100 customers in one go. Commercial and open source solver can be considered as the enumerated inverted tree and size of the tree and computational time increases with the size of the problem. An alternative way would be to get the optimal or near-optimal solution by integrating the clustering method with the exact method. If we increase the number of customers in each cluster, then we can expect to get an optimal solution instead of a sub-optimal solution (Yüce *et* al., 2018).

2.1 K-means and Gaussian Mixture:

Clustering, an unsupervised learning, partitions the observations into distinct groups so that observations within each group are similar to each other to achieve high intra-cluster similarity and observations in different groups are different from each other to achieve low inter-cluster similarity (Hastie *et* al., 2017, pp 385, Nasiriany *et* al., 2019, pp 152). K-means, a centroid-based clustering, is a variant of expectation maximization (EM) algorithm for Gaussian Mixture Model (GMM) or Mixture of Gaussians (GOM) (Murphy, 2012, pp 352).

If Y denotes the set of N data points $y_i \in \mathbb{R}^d \, and \, Y = C_1 \cup C_2 \cup C_3 \ldots \ldots \cup C_k$ where C_k is the k[th] cluster. In the centroid-based clustering, cluster C_k can be represented by a point $c_k \in \mathbb{R}^d$. K-means minimizes the below objective function:

$$\arg min_{Y=C_1 \cup C_2 \ldots \cup C_k} \sum_{k=1}^{K} \sum_{y \in C_k} \|y - c_k\|^2$$

and the centroid of the cluster, c_k^* , can be obtained by minimizing the sum-squared-error as follows:

$$c_k^* = \arg min_{c_k} \sum_{y \in C_k} \|y - c_k\|^2 \approx \frac{1}{|C_k|} \sum\nolimits_{y \in C_k} y$$

In K-means, equal priority is given to all features and 2d projection of the cluster, thus, it will look like a circle. The literature review shows that k-means, k-medoids, DBSCAN, recursive DBSCAN etc have been used to cluster data in vehicle route optimization problems (Cömert *et* al., 2017).

The Latent Variable Model (LVM) is a probabilistic model. In LVM, some variables can be measured directly and some variables remain hidden or latent. Only the effect of hidden variables can be measured on the observed variables (Nasiriany *et* al., 2019 pp 156). GMM is basically an LVM.

If $\boldsymbol{\mu}_k$ is the mean of Gaussian multivariate and $\sum_k$ is the covariance matrix then GMM model can be represented as $p(x_i \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathrm{N}(x_i \mid \boldsymbol{\mu}_k, \sum_k)$ where π_k is the mixing weight and $\sum_{k=1}^{K} \pi_k = 1 \; and \; 0 \leq \pi_k \leq 1$. If z_i is the latent variable for data point i then posterior probability $p(z_i = k \mid x_i, \boldsymbol{\theta})$ refers to the probability that data point i belongs to cluster k. This is also known as responsibility of cluster k for data point i, r_{ik} (Murphy, 2012, pp 340).

2.2 Autoencoder:

An autoencoder is a generalized version of Principal Component Analysis (PCA) (Bank et. *al*, 2021). It is a specific type of neural network which consists of an encoder and a decoder. The encoder compresses the input data to create a meaningful representation and the decoder uses the same representation to return the original input. If X is the encoder then X: $\mathbb{R}^n \rightarrow \mathbb{R}^p$. If Y is the decoder then Y: $\mathbb{R}^p \rightarrow \mathbb{R}^n$ to satisfy the below:

$$arg\ min_{X,Y} = E[l(x, Y \circ X(x)]$$

where E is the expectation over distribution x, and l is the reconstruction loss which is basically a $l_2 - norm \ (Baldi, 2012)$. K-means and other unsupervised clustering methods are sensitive to the dimension of input data. Usually, the output of the encoder, the latent representation, is taken as input for K-means (Bank et. *al*, 2021).

3. Result and Discussions:

Validated the result obtained from the two-indexed BigM model, stated in Sec.2, for the 'C' and 'R' series of Solomon's VRPTW instances and used the same model for further study. Feature selection can be broadly classified into four categories, namely, filters, wrappers, hybrid, and embedded. The filter method selects a subset of features by evaluating statistical performance of features such as Pearson correlation coefficient, maximal information compression index (MICI) etc. The filter method was used to select features and a subset of selected features was used further to reduce the computational burden, shown in table 1.

Table 1 Selected features

Method	Selected Features	Selected Subset
Silhouette Score	'DEMAND', 'YCOORD', 'XCOORD', 'READY_TIME', 'DUE_DATE'	'XCOORD','YCOORD','DEMAND',

To determine the number of clusters, Distortion Score (DS), Silhouette Score (SS), and Caliniski Harabasz Score (CHS) were used, shown in fig 2. Both DS and CHS predicted 6 clusters and SS predicted 2 clusters. Rejected SS as large number of data points will increase computational time and MIP gap. A baseline K-means model was prepared with 6 clusters for further study.

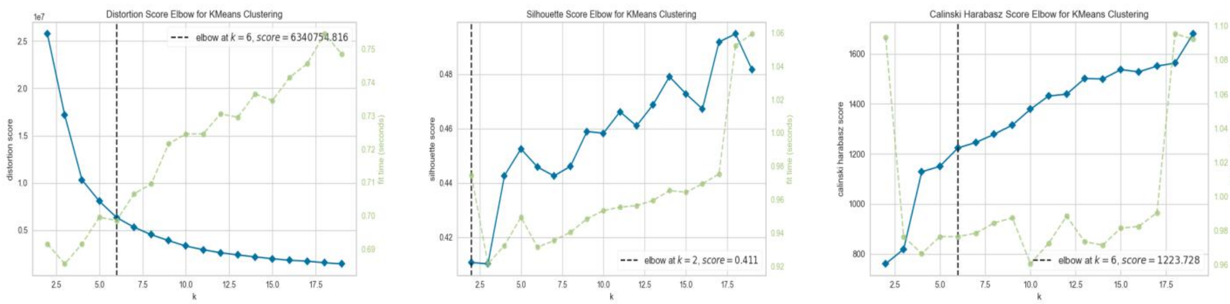


Fig 2 Selecting number of cluster using Distortion score, Silhouette score, and Calinski Harabasz score

Considering 6 and 8 clusters further to study the effect of each cluster on the 1000 VRPTW problem, shown in table 2 and 3. A two-indexed BigM model was used for each clustered data to generate an .lp file, mentioned as "Layer". A layer is a computer generated mathematical model for a given input data to yield feasible routes. The obtained result was compared with the published result. A study reveals that 8 clusters take 7447 secs i.e. 124.11 mins, and yielded more than 1% error in assigning customers to the wrong route as per the published results (Quintiq, 2015). The obtained MIP gap is 0.0% and 101 vehicles are required. 6 clusters yield less than 1% error with a 0.0% MIP gap to assign customers to the wrong route as per the published result, take 8423 secs i.e.140.38 mins, and 100 vehicles are required. If we fix the MIP Gap at 5.0% then computational time of Layer 3 for 6 clusters would be 14 secs and total computational time would be 109 secs i.e. 1.82 mins. It can be further reduced using parallel computation. The given input dataset has no target column. Therefore, table 3 was used to compute the target column or labels, considering zero errors.

Table 2 Effect of 8 clusters on 1000 VRPTW instance

Layer-0 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 2.00 Solving Nodes : 1 Primal Bound : +1.15918000000000e+03 (4 solutions) Dual Bound : +1.15918000000000e+03 Gap : 0.00 %	**Total:8 vehicles** Route 5: 0->130->725->612->904->855->13->78->794->971->628->376->817->0 Route 13: 0->183->36->663->714->561->498->438->22->311->893->266->0 Route 4: 0->473->109->77->417->424->90->395->519->728->65->186->0 Route 19: 0->488->800->211->359->442->804->715->888->923->355->0 Route 15: 0->544->278->511->308->191->712->898->979->915->732->0 Route 9: 0->581->933->52->335->368->654->918->530->444->699->59->140->0 Route 27: 0->615->667->808->926->964->988->277->286->824->0 Route 10: 0->968->745->218->371->643->143->844->956->661->0
Layer -1	**Total: 10 vehicles**

SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 2.00 Solving Nodes : 1 Primal Bound : +5.15453000000000e+03 (4 solutions) Dual Bound : +5.15453000000000e+03 Gap : 0.00 %	Route 45: 0->6->268->980->210->574->118->897->202->547->0 Route 37: 0->28->775->973->188->527->245->793->636->147->747->0 Route 32: 0->115->962->486->743->440->512->397->939->123->0 Route 44: 0->363->797->45->68->575->295->180->302->187->19->0 Route 39: 0->451->605->578->346->443->7->206->427->67->685->0 Route 47: 0->482->625->990->584->60->809->993->864->790->280->43->0 Route 36: 0->785->267->837->104->132->925->526->223->24->217->568->0 Route 54: 0->849->476->788->528->494->354->555->342->558->549>552->0 Route 22: 0->914->362->621->969->213->251->455->252->86->485->0 Route 48: 0->960->810->620->959->841->580->304->480->684->833->56->0
Layer -2 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 10.00 Solving Nodes : 1 Primal Bound : +8.23814000000000e+03 (19 solutions) Dual Bound : +8.23814000000000e+03 Gap : 0.00 %	**Total: 15 vehicles** Route 79: 0->35->825->932->868->154->370->827->832->270->171->146->0 Route 82: 0->54->483->531->607->177->195->769->135->724->379->525->0 Route 73: 0->108->182->470->93->203->716->706->875->693->762->601->232->0 **Route 85: 0->151->137->998->163->26->665->0 (Actual Route 85 : 931 173 151 137 998 163 26 665 340) **Route 90: 0->164->323->242->419->179->622->450->0 (Actual Route 90 : 164 323 242 419 179 622 450 540) Route 94: 0->175->144->373->692->133->913->539->891->892->507->0 Route 83: 0->184->139->955->14->645->754->700->889->791->0 Route 92: 0->194->84->867->779->908->595->468->847->916->0 Route 91: 0->299->449->602->546-.>759->377->296->995->0 Route 88: 0->384->853->432->727->288->835->858->361->839->608->994->447->0 Route 84: 0->582->479->437->719->204->37->292->696->205->746->514->0 Route 86: 0->722->718->326->334->100->414->430->802->550->0 Route 89: 0->780->529->300->965->740->677->921->989->917->0 Route 87: 0->957->787->806->710->318->831->961->587->58->290->0 Route 98: 0->982->707->111->551->94->843->942->885->609->877->753->0
Layer -3 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 4.00 Solving Nodes : 1 Primal Bound : +4.31541000000000e+03 (4 solutions) Dual Bound : +4.31541000000000e+03 Gap : 0.00 %	**Total: 12 vehicles** Route 16: 0->66->545->778->129->8->491->821->150->383->695->297->0 Route 55: 0->116->813->840->463->586->906->541->886->174->522->0 Route 57: 0->285->356->472->393->967->474->688->315->518->946->0 Route 53: 0->380->44->4->17->9->89->18->105->97->919->666->0 Route 18: 0->394->441->40->282->298->119->610->23->524->0 Route 56: 0->411->563->986->938->818->910->27->189->680->153->0 Route 7: 0->577->533->329->249->945->20->46->689->138->905->225->0 Route 21: 0->819->999->246->943->723->920->227->850->828->433->567->161->71->862->0 Route 17: 0->902->543->679->604->742->781->880->703->493->0 Route 30: 0->922->736->34->912->61->744->911->811->0 Route 58: 0->937->181->407->102->934->720->126->265->2->53->62->0 Route 6: 0->981->125->954->404->39->782->256->857->510->0
Layer – 4 SCIP Status : problem	**Total: 15 vehicles** Route 61: 0->95->834->557->381->271->215->630->80->236->264->750->642->310->0

is solved [optimal solution found] Solving Time (sec) : 7405.00 Solving Nodes : 1 (total of 2 nodes in 2 runs) Primal Bound : +7.71782000000000e+03 (57 solutions) Dual Bound : +7.71782000000000e+03 Gap : 0.00 %	Route 69: 0->122->705->316->678->461->47->773->389->603->0 Route 65: 0->190->842->50->675->822->459->830->872->903->0 Route 66: 0->301->777->564->320->360->763->641->284->878->0 Route 71: 0->351->127->682->475->235->596->749->429->0 Route 68: 0->382->784->708->460->852->428->907->0 Route 63: 0->406->820->502->222->927->509->465->196->674->871->662->0 Route 62: 0->415->322->627->462->698->559->172->569->0 Route 60: 0->532->33->209->709->991->372->894->672->805->0 Route 64: 0->536->257->121->966->635->247->113->452->448->367->347->83->500->0 Route 24: 0->562->148->291->57->131->593->953->560->418->159->513->374->0 Route 70: 0->616->313->481->74->876->795->114->338->895->416->234->0 **Route 75: 0->846->634->624->201->445->517->454->0 (Actual Route 75 : 328 92 200 846 634 624 201 445 517 454) Route 59: 0->977->390->352->600->870->786->274->869->0 Route 67: 0->984->896->244->497->48->935->484->653->975->434->765->0
Layer -5 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 3.00 Solving Nodes : 1 Primal Bound : +6.11264000000000e+03 (8 solutions) Dual Bound : +6.11264000000000e+03 Gap : 0.00 %	**Total: 12 vehicles** Route 38: 0->79->664->49->489->490->16->669->731->375->701->436->387->0 Route 41: 0->106->158->81->385->537->124->816->412->160->0 Route 50: 0->208->425->149->332->345->936->739->409->0 Route 28: 0->238->597->287->772->506->626->471->260->0 Route 42: 0->364->492->219->170->240->216->349->178->237->176->0 Route 49: 0->535->303->226->671->978->851->134->358->0 Route 46: 0->570->350->882->400->752->269->865->75->713->185->0 Route 40: 0->571->681->948->91->96->378->275->341->431->606->0 Route 52: 0->652->823->258->317->958->505->142->516->0 Route 51: 0->711->403->970->729->413->866->887->408->690->812->614->845->0 Route 43: 0->789->553->103->331->972->974->422->944->401->579->538->0 Route 26: 0->815->658->656->771->761->168->592->464->281->391->0
Layer – 6 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 18.00 Solving Nodes : 1 Primal Bound : +5.54748000000000e+03 (10 solutions) Dual Bound : +5.54748000000000e+03 Gap : 0.00 %	**Total: 17 vehicles** Route 100: 0->167->293->542->928->952->619->446->15->73->0 Route 3: 0->231->1->70->496->166->515->307->687->676->0 Route 99 : 0->309->890->11->721->253->651->396->733->741->469->0 Route 97: 0->336->508->254->152->112->157->29->757->101->392->453->860->283->221->0 Route 96: 0->435->250->629->829->523->117->768->657->863->330->248->826->0 Route 93: 0->487->255->899->289->572->950->423->85->554->233->881->650->0 Route 12: 0->599->755->704->99->10->649->556->598->273->717->0 Route 72: 0->611->940->534->738->807->314->770->617->0 Route 34: 0->623->734->228->107->776->162->992->583->0 Route 29: 0->638->72->573->337->613->947->588->439->751->0 Route 35: 0->655->976->458->798->51->591->141->239->220->55->883->637->848->647->0 Route 1: 0->660->3->402->456->565->193->670->646->263->207->0 Route 11: 0->836->69->694->796->951->495->420->566->21->0 Route 95: 0->854->783->640->861->339->457->748->325->673->735->0 Route 2: 0->924->155->224->590->312->702->199->737->873->576->0

	Route 33: 0->963->726->801->987->466->279->31->276->41->0 Route 23: 0->997->87->585->365->120->521->996->0
Layer -7 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 3.00 Solving Nodes : 1 Primal Bound : +5.14384000000000e+03 (4 solutions) Dual Bound : +5.14384000000000e+03 Gap : 0.00 %	**Total: 12 vehicles** Route 74: 0->76->631->467->333->639->229->730->88->659->110->42->0 Route 80: 0->38->633->262->63->618->632->366->192->12->929->0 Route 77: 0->212->859->324->421->369->261->838->697->814->0 **Route 31: 0->306->294->504->766->343->930->756->241->230->169->540->0 (Actual Route 31 : 306 294 504 766 343 930 756 241 230 169) **Route 75: 0->328->92->200->426->799->594->503->399->98->82->156->0 (Actual Route 75 : 328 92 200 846 634 624 201 445 517 454) Route 81: 0->386->792->901->909->259->319->145->803->165->128->691->668->874->0 Route 14: 0->388->760->272->321->353->983->499->405->548->0 Route 78: 0->648->949->197->767->214->198->64->589->0 **Route 76: 0->764->879->501->683->344->0 (Actual Route 76: 764 879 501 683 344 426 799 594 503 399 98 82 156) Route 25: 0->856->758->774->478->410->5->477->398->0 **Route 85: 0->931->173->340->0 (Actual Route 85: 931 173 151 137 998 163 26 665 340) Route 8: 0->941->32->520->357->686->243->327->348->0 Route 20: 0->985->884->305->30->136->900->644->25->1000->0

- Error = $\left(\sum customers\ who\ are\ assigned\ to\ wrong\ route/1000\right) * 100\%$, Accuracy = 1-Error
- MIP Gap = $\frac{|primal-dual|}{\min(|primal|,|dual|)} \times 100$

Table 3 Effect of 6 clusters on 1000 VRPTW instance

Layer -0 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 3.00 Solving Nodes : 1 Primal Bound : +5.48336000000000e+03 (4 solutions) Dual Bound : +5.48336000000000e+03 Gap : 0.00 %	**Total : 11 vehicles** Route #45: 0->6->268->980->210->574->118->897->202->547->0 Route #37: 0->28->775->973->188->527->245->793->636->147->747->0 Route #32: 0->115->962->486->743->440->512->397->939->123->0 Route #28 : 0->238->597->287->772->506->626->471->260->0 Route #44: 0->363->797->45->68->575->295->180->302->187->19->0 Route #39: 0->451->605->578->346->443->7->206->427->67->685->0 Route #47: 0->482->625->990->584->60->809->993->864->790->280->43->0 Route #36: 0->785->267->837->104->132->925->526->223->24->217->568->0 Route #54: 0->849->476->788->528->494->354->555->342->558->549->552->0 Route #22: 0->914->362->621->969->213->251->455->252->86->485->0 Route #48: 0->960->810->620->959->841->580->304->480->684->833->56->0
Layer -1 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 11.00 Solving Nodes : 1 Primal Bound :	**Total : 16 Vehicles** Route #16: 0->66->545->778->129->8->491->821->150->383->695->297->0 **Route #55 : 0->116->813->840->463->586->906->541->886->174->522->811->0 (Actual Route 55 : 116 813 840 463 586 906 541 886 174 522) Route #5: 0->130->725->612->904->855->13->78->794->971->628->376->817->0 Route #13: 0->183->36->663->714->561->498->438->22->311->893->266->0 Route #18 : 0->394->441->40->282->298->119->610->23->524->0

+3.65448000000000e+03 (9 solutions) Dual Bound : +3.65448000000000e+03 Gap : 0.00 %	Route #56: 0->411->563->986->938->818->910->27->189->680->153->0 Route #4: 0->473->109->77->417->424->90->395->519->728->65->186->0 Route #19: 0->488->800->211->359->442->804->715->888->923->355->0 Route #15: 0->544->278->511->308->191->712->898->979->915->732->0 **Route #7: 0->577->533->329->249->945->20->46->689->138->905->225->946->0 (Actual Route 7 : 577 533 329 249 945 20 46 689 138 905 225) Route #9: 0->581->933->52->335->368->654->918->530->444->699->59->140->0 Route #27: 0->615->667->808->926->964->988->277->286->824->0 Route #21: 0->819->999->246->943->723->920->227->850->828->433->567->161->71->862->0 Route #17: 0->902->543->679->604->742->781->880->703->493->0 Route #10: 0->968->745->218->371->643->143->844->956->661->0 Route #6: 0->981->125->954->404->39->782->256->857->510->0
Layer -2 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 4.00 Solving Nodes : 1 Primal Bound : +7.61184000000000e+03 (8 solutions) Dual Bound : +7.61184000000000e+03 Gap : 0.00 %	**Total :15 vehicles** Route #38: 0->79->664->49->489->490->16->669->731->375->701->436->387->0 Route #41: 0->106->158->81->385->537->124->816->412->160->0 Route #50: 0->208->425->149->332->345->936->739->409->0 **Route #57: 0->285->356->472->393->967->474->688->315->518->0 (Actual Route 57 : 285 356 472 393 967 474 688 315 518 946) Route #42: 0->364->492->219->170->240->216->349->178->237->176->0 Route #53: 0->380->44->4->17->9->89->18->105->97->919->666->0 Route #49: 0->535->303->226->671->978->851->134->358->0 Route #46: 0->570->350->882->400->752->269->865->75->713->185->0 Route #40: 0->571->681->948->91->96->378->275->341->431->606->0 Route #52: 0->652->823->258->317->958->505->142->516->0 Route #51: 0->711->403->970->729->413->866->887->408->690->812->614->845->0 Route #43: 0->789->553->103->331->972->974->422->944->401->579->538->0 Route #26: 0->815->658->656->771->761->168->592->464->281->391->0 **Route #30: 0->922->736->34->912->61->744->911->0 (Actual Route 30: 922 736 34 912 61 744 911 811) Route #58: 0->937->181->407->102->934->720->126->265->2->53->62->0
Layer – 3 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 8328.00 Solving Nodes : 1 (total of 2 nodes in 2 runs) Primal Bound : +8.66187999999998e+03 (52 solutions) Dual Bound : +8.66187999999998e+03 Gap : 0.00 %	**Total : 17 vehicles** Route #61: 0->95->834->557->381->271->215->630->80->236->264->750->642->310->0 Route #69: 0->122->705->316->678->461->47->773->389->603->0 Route #65: 0->190->842->50->675->822->459->830->872->903->0 Route #66: 0->301->777->564->320->360->763->641->284->878->0 Route #75: 0->328->92->200->846->634->624->201->445->517->454->0 Route #71: 0->351->127->682->475->235->596->749->429->0 Route #68: 0->382->784->708->460->852->428->907->0 Route #63 : 0->406->820->502->222->927->509->465->196->674->871->662->0 Route #62: 0->415->322->627->462->698->559->172->569->0 Route #60: 0->532->33->209->709->991->372->894->672->805->0 Route #64: 0->536->257->121->966->635->247->113->452->448->367->347->83->500->0 Route #24: 0->562->148->291->57->131->593->953->560->418->159->513->374->0 Route #70: 0->616->313->481->74->876->795->114->338->895->416->234->0

	Route #78: 0->648->949->197->767->214->198->64->589->0 Route #78: 0->764->879->501->683->344->426->799->594->503->399->98->82->156->0 Route #59: 0->977->390->352->600->870->786->274->869->0 Route #67: 0->984->896->244->497->48->935->484->653->975->434->765->0
Layer -4 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 23.00 Solving Nodes : 1 Primal Bound : +6.52484000000000e+03 (10 solutions) Dual Bound : +6.52484000000000e+03 Gap : 0.00 %	**Total : 19 vehicles** Route #100: 0->167->293->542->928->952->619->446->15->73->0 Route #92: 0->194->84->867->779->908->595->468->847->916->0 Route #3: 0->231->1->70->496->166->515->307->687->676->0 Route #91: 0->299->449->602->546->759->377->296->995->0 Route #99: 0->309->890->11->721->253->651->396->733->741->469->0 Route #97: 0->336->508->254->152->112->157->29->757->101->392->453->860->283->221->0 Route #96: 0->435->250->629->829->523->117->768->657->863->330->248->826->0 Route #93: 0->487->255->899->289->572->950->423->85->554->233->881->650->0 Route #12: 0->599->755->704->99->10->649->556->598->273->717->0 Route #72: 0->611->940->534->738->807->314->770->617->0 Result #34: 0->623->734->228->107->776->162->992->583->0 Route #29: 0->638->72->573->337->613->947->588->439->751->0 Route #35: 0->655->976->458->798->51->591->141->239->220->55->883->637->848->647->0 Route #1: 0->660->3->402->456->565->193->670->646->263->207->0 Route #11: 0->836->69->694->796->951->495->420->566->21->0 Route #95: 0->854->783->640->861->339->457->748->325->673->735->0 Route #2: 0->924->155->224->590->312->702->199->737->873->576->0 Route #33: 0->963->726->801->987->466->279->31->276->41->0 Route #23: 0->997->87->585->365->120->521->996->0
Layer -5 SCIP Status : problem is solved [optimal solution found] Solving Time (sec) : 54.00 Solving Nodes : 1 Primal Bound : +1.05505400000000e+04 (45 solutions) Dual Bound : +1.05505400000000e+04 Gap : 0.00 %	**Total :22 vehicles** Result #79: 0->35->825->932->868->154->370->827->832->270->171->146->0 Route #80: 0->38->633->262->63->618->632->366->192->12->929->0 Route #82: 0->54->483->531->607->177->195->769->135->724->379->525->0 Route #74: 0->76->631->467->333->639->229->730->88->659->110->42->0 Route #73: 0->108->182->470->93->203->716->706->875->693->762->601->232->0 Route #90: 0->164->323->242->419->179->622->450->540->0 Route #94: 0->175->144->373->692->133->913->539->891->892->507->0 Route #83: 0->184->139->955->14->645->754->700->889->791->0 Route #77: 0->212->859->324->421->369->261->838->697->814->0 Route #31: 0->306->294->504->766->343->930->756->241->230->169->0 Route #88: 0->384->853->432->727->288->835->858->361->839->608->994->447->0 Route #81: 0->386->792->901->909->259->319->145->803->165->128->691->668->874->0 Route #14: 0->388->760->272->321->353->983->499->405->548->0 Route #84: 0->582->479->437->719->204->37->292->696->205->746->514->0 Route #86: 0->722->718->326->334->100->414->430->802->550->0 Route #89: 0->780->529->300->965->740->677->921->989->917->0 Route #25: 0->856->758->774->478->410->5->477->398->0 Route #85: 0->931->173->151->137->998->163->26->665->340->0

	Route #8: 0->941->32->520->357->686->243->327->348->0 Route #87: 0->957->787->806->710->318->831->961->587->58->290->0 Route #98: 0->982->707->111->551->94->843->942->885->609->877->753->0 Route #20: 0->985->884->305->30->136->900->644->25->1000->0

- Error = $\left(\sum customers\ who\ are\ assigned\ to\ wrong\ route/1000\right) * 100\%$, Accuracy = 1-Error
- MIP Gap = $\frac{|primal-dual|}{\min(|primal|,|dual|)} \times 100$

GMM model uses spherical, tied, diag, and full covariance. Selected tied covariance and 6 components for GMM model, shown in fig 3 and 4.

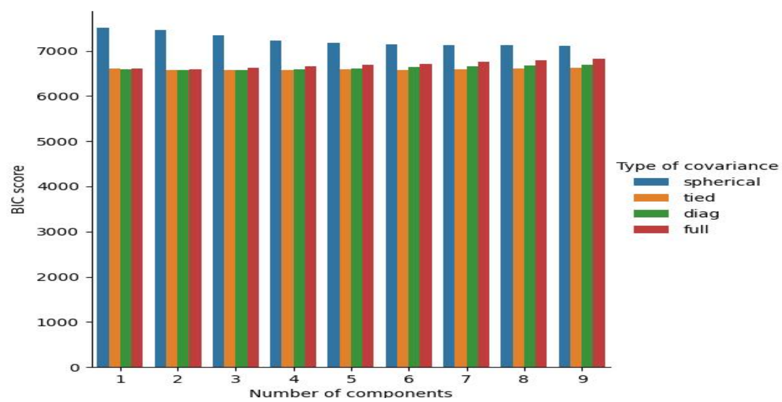
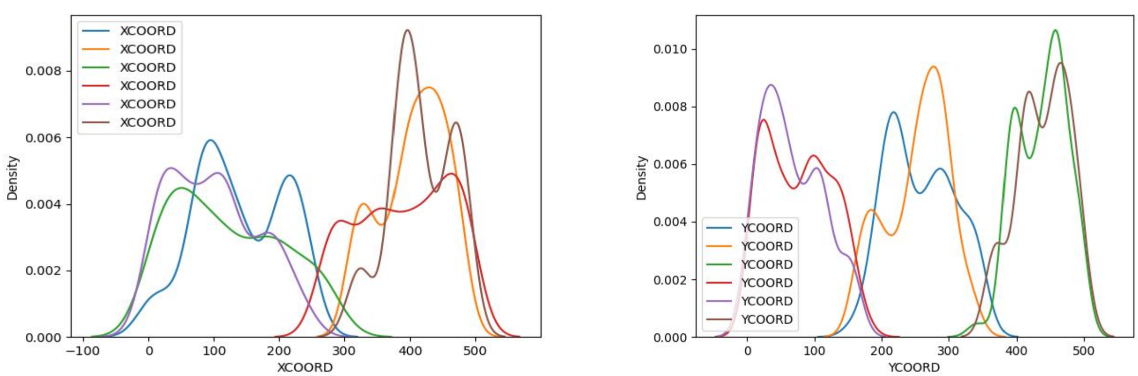


Fig 3 Selection of GMM components using BIC score for different type of covariance



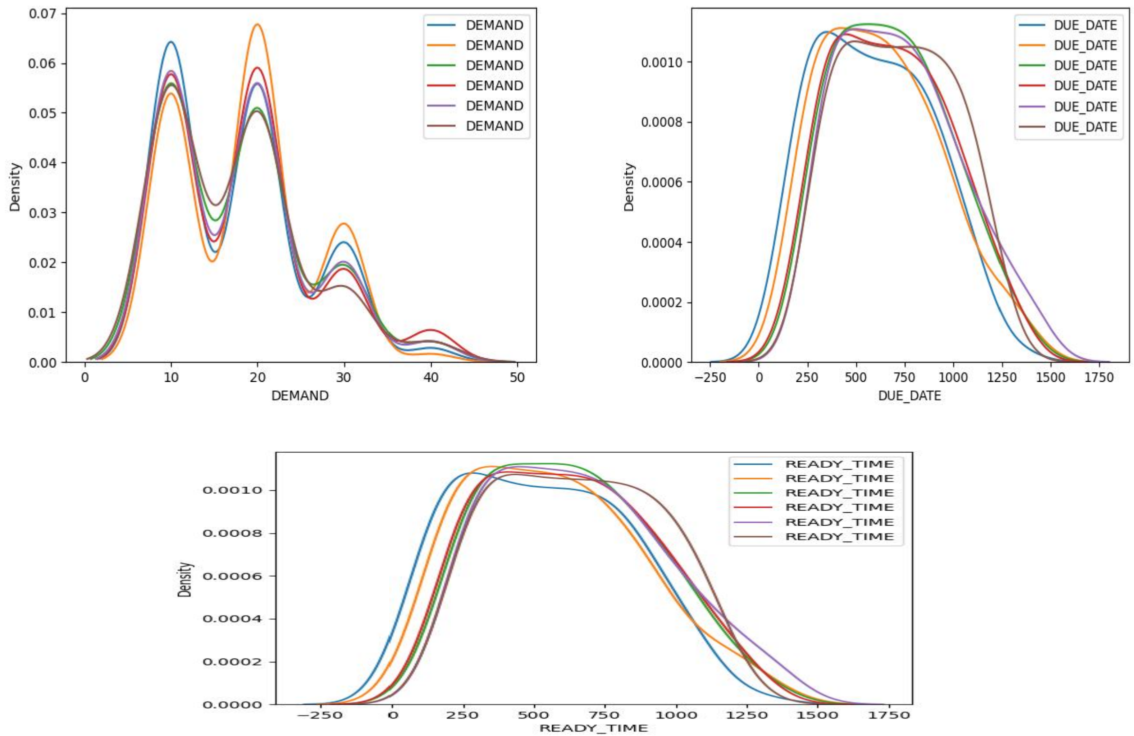


Fig 4 Kernel density estimation plot of different dimensions of 1000 VRPTW dataset

A deep autoencoder model was built by tuning batch size, epochs, activation function, and optimizer to select the best model of autoencoder, shown in table 4. Obtained MSE for the best model is shown table 5.

Table 4 Selecting parameters of autoencoder

Activation Function	Optimizer	MSE	
		Training	Testing
Linear	SGD	0.4428	0.4213
Relu	SGD	0.2309	0.2579
Relu	Adam	0.1138	0.1310
relu with kernel constraint	Adam	5.7204	5.9276

Table 5 MSE in training and testing of autoencoder

	MSE
Training	0.0360
Testing	0.0376

K-means outperform GMM and integrated use of autoencoder and K-means, shown in table 6. The proposed approach can give near-optimal solution with accurate number of vechicles (i.e. 100).

Table 6 Comparison of clustering performance in terms of accuracy (%)

Models	ACC	NMI	ARI
K-means	99.8	99.3	99.5
GMM	93.5	89.7	88.2
Autoencoder+K-means	92.0	85.2	84.1

- NMI : Normalized Mutual Information; ACC: Clustering Accuracy; ARI : Adjusted Rand Index

For real-time GPS dataset, we need to use map matching algorithm to calculate exact distance. Modified distance, c_{ij}, will be used in the mathematical model proposed in sec 2. Average velocity of each vehicle, v_{ij}, will be calculated from real-time data further to modify the eq. 4 as follows:

$$t_i + STime_i + \frac{c_{i,j}}{v_{ij}} + BigM * X_{i,j} + w_j \leq t_j + BigM \quad \forall\, i,j \in A\,, i \neq j \; and \; j \neq 0 \ldots\ldots (4)$$

4. Conclusions

This paper shows that the wrong clustering approach increases subtour, MIP Gap, and computational time. No researchers have ever shown the effect of clustering on subtour and MIP Gap. In the context of BigData, K-means is highly effective for streaming data and it is also available in Spark ML. This computational study shows that K-means $\succ GMM \succ Integrated\ Autoencoder\ K - means$. One of the most significant contribution of this paper is the concept of "Layer", the most effective method, to yield near-optimal solution for 1000 VRPTW problem in less than two minutes. The originator of this problem used two stages metaheuristic not the exact method.

References:

1. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. Retrieve from http://cepac.cheme.cmu.edu/pasi2011/library/cerda/braysy-gendreau-vrp-review.pdf
2. Farid Ghannadpour S. and Hooshfar M.(2015). Multi-objective Evolutionary Method for Dynamic Vehicle Routing and Scheduling Problem with Customers' Satisfaction Level. In *Proceedings of the International Conference on Operations Research and Enterprise Systems* (ICORES-2015), pages 91-98.DOI: 10.5220/0005172600910098
3. M. Barkaoui , J. Berger , A. Boukhtouta (2015). Customer satisfaction in dynamic vehicle routing problem with time windows, Applied Soft Computing, Vol. 35, pp.423-432.

4. Nathalie De Jaegere, Mieke Defraeyea and Inneke Van Nieuwenhuyse (2014). The Vehicle Routing Problem: State of the Art Classification and Review. Retrieved from https://core.ac.uk/download/pdf/34605432.pdf
5. Jing Fang (2011). The Vehicle Routing Problem with Simultaneous Pickup and Delivery Based on Customer Satisfaction, Vol.15, pp. 5284-5289.
6. Mehmet Fatih Yüce, Ali Gunes, Metin Zontul, Tuğba Altintas. Time Window and Location Based Clustered Routing with Big and Distributed Data. Industrial Engineering. Vol. 2, No. 2, 2018, pp. 42-51. doi: 10.11648/j.ie.20180202.11
7. Serap Ercan Cömert, Harun Reşit Yazgan, Irem Sertvuran and Hanife Şengǜl (2017). A n e w approach for solution of vehicle routing problem with hard time window : an application in a super market chain, Sadhana, Vol. 42, No. 12, pp. 2067–2080.
8. Ropke, Stefan, and David Pisinger. 2006. "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows." *Transportation Science* 40 (4): 455–472.
9. Bai, Ruibin, Edmund K. Burke, Michel Gendreau, Graham Kendall, and Barry McCollum. 2007. *Memory length in hyper-heuristics: An empirical study*. New York: Ieee
10. Hanjun Dai, Bo Dai, and Le Song. "Discriminative Embeddings of Latent Variable Models for Structured Data". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 2702–2711.
11. Mohammadreza Nazari et al. *Reinforcement Learning for Solving the Vehicle Routing Problem*. 2018. arXiv: 1802.04240 [cs.AI].
12. Hottung, André, and Kevin Tierney. 2019. "Neural large neighborhood search for the capacitated vehicle routing problem." In *arXiv*, https://arxiv.org/abs/1911.09539.
13. Wouter Kool, Herke van Hoof, and Max Welling. "Attention, Learn to Solve Routing Problems!" In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
14. Maryam Abdirad & Krishna Krishnan. Industry 4.0 in Logistics and Supply Chain Management: A Systematic Literature Review, Engineering Management Journal,2020.
15. Battarra, M., Erdoğan, G., and Vigo, D., Exact Algorithms for the Clustered Vehicle Routing Problem, *Operations Research,*
16. Barreto, S., Ferreira, S., Paixão, J., and Santos, S. B. Using clustering analysis in a capacitated location-routing problem, European Journal of Operational Research, Vol.179, pp. 968-977,2007.
17. Alesiani, F., Ermis, G., and Gkiotsalitis, K. Constrained Clustering for the Capacitated Vehicle Routing Problem (CC-CVRP), *Applied Artificial Intelligence,* Vol.36,No.1, 2022.
18. Abbas, F., and Fan,P. Clustering-based reliable low-latency routing scheme using ACO method for vehicular networks, *Vehicular Communications*,Vol.12, pp.66-74,2018.
19. Song, C., Liu, F., Huang, Y., Wang,L., and Tan, T. Auto-encoder Based Data Clustering,2013.
20. Sahal, R., Breslin, J.G., Ali, M.,I. (2020) Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case, Journal of Manufacturing Systems, Vol.54,pp.138-151.
21. Bellavista, P., Bosi, F., Corradi, A., Foschini,L., Monti, S., Patera, L., Poli, L., Scotece,D., and Solimando, M. (2019) Design Guidelines for Big Data Gathering in Industry 4.0 Environments

22. Peres, R.S., Rocha, A.D., Coelho, A., and Barata, J. A Highly Flexible, Distributed Data Analysis Framework for Industry 4.0 Manufacturing Systems
23. Gareth, J., Witten, D., Hastie, T., and Tibshirani, R. (2017) An Introduction to Statistical Learning with Applications in R, ISBN 978-1-4614-7138-7 (eBook).
24. Nasiriany, S., Thomas, G., Wang, W., Yang, A., Listgarten, J., Sahai, A. (2019) A Comprehensive Guide to Machine Learning.
25. Murphy, K.P. (2012) Machine Learning A Probabilistic Perspective, MIT Press, England.
26. Quintiq. (2015). https://www.sintef.no/contentassets/51a833740c45438b99b2935fd1c057d1/c1_10_1.42478.95.sintef.txt