Special Student Project

Massive Data Visual Analytic with Neural Network

Author:
Zhenghao Chen

Supervisor:
Dr. Xiuying Wang
Dr. Jianlong Zhou

July 19, 2016

Abstract

Faced problems with massive data, researchers came up with a number of efficient methodologies including classification, clustering and regression. In this study, a methodology named Artificial Neural Network (ANN) will be concerned and used to solve the data mining problems. Introduction part will explain the algorithm and architecture of basic neural network and its resources. Also, the project will cover 2 big problems and each problem will consists of 2 parts. First problem is a study process that to solve the classical classification problem that specifically text and image classification using Deep Multi-layers Neural Network (DNN) and Convolution Neural Network (CNN) respectively. While the second problem is my own contribution that using visual analytic technique to build up a neural net back-end system to resolve the security and management issue in a theme park Dino-fun world. For that system, Self-Organizing Map (SOM) will be used to perform clustering problem for detecting anomalies patterns as well as Recurrent Neural Network (RNN) is implemented for traffic flow prediction. In the future work and conclusion part, I will summary the advantages and usability of Neural Network and illustrate why we should use this technique for further problems.

1 Introduction

For past couple of decades, Artificial Neural Network (ANN) serves as a decent machine learning technique to deal with massive data problem in many industries for either supervised learning or unsupervised learning. It has a long history from 1940s [25], however as its complicated computational complex and sophisticated implementation, it was more likely to be an ideal model rather to be implemented in real industry in early time. In recent years, thanks to the availability of inexpensive, parallel hardware (GPUs, computer clusters) and massive amounts of data, neural network technique became much more successful methodology in machine learning [16] especially for image recognition, natural language processing and massive data classification and clustering etc. Deep learning [16] is a machine learning model based on multi-layers neural network, specifically, as key characteristic of a neural network is its ability to learn, therefore we are able to use different multi-layers network architecture for wide variety of problems and try to build a comprehensive model to illustrate the relationship, pattern and so on from given data after learning rather than only building a convenient mathematical model like inference model [20]. A lot of useful networks has been mathematically proven that can be implemented efficiently in some particular problems. For instance, Multi-layers Convolution Neural Network (CNN) can be used for image classification like hand-writing digit recognition [15] while Recurrent Neural Network (RNN) could have outstanding performance in sequence data like Natural Language Processing [6] and time-series trajectory data. The aim of this study is to have deep understanding of neural network and implement this techniques to solve the data mining problem. The first problem is a classification problem that classify text data which is UCI digit letter data[1] using Deep Multi-layers Neural Net and classify image data MNIST hand writing digit [2] using convolution neural net. The second problem is 2015 Visual Analytics Science and Technology Challenge problem(VAST 2015) [3] in this problem, I will use Self-Organizing Map to cluster the visiting patterns of customers in a theme park in order to find the abnormal pattern to address the crime issue and Recurrent Neural Network to predict the traffic flow in different time-stamp for security management in that park. Overall, by showing how to use neural network in those problems and evaluate them by comparing with other classical methodologies like Support Vector Machine (SVM), Random Forest(RF) etc it would be more straightforward to see the advantages of neural approach and the reason why we should use them for further problems.

2 Neural Network

In this section, I will briefly introduce the neural network that its architecture and basic algorithm. That is for one, I will start from neuron, followed by layer and the whole architecture. For another, the basic algorithm that forward activation and backward propagation will be explained.

[1] `http://archive.ics.uci.edu/ml/datasets/Letter+Recognition`
[2] `http://yann.lecun.com/exdb/mnist/`
[3] `http://vacommunity.org/VAST+Challenge+2015`

2.1 Architecture

2.1.1 Neuron

Neural Network is constructed with basic neuron element. Figure 1 shows that basic structure of single neuron. There are 4 basic elements associated with each neuron that are weights, activation function, input and output. The weight is associated with input, it could be a singular value, vector or any particular structure depends on category of neural network for instance in convolution neural network the weight are represented by convolution filter [20] [21]. Sum of product of each weight and input will be processed with activation function that I will explain how it works in activation part. After that output that produced by activation function will be taken as one of input to neuron in next layer.

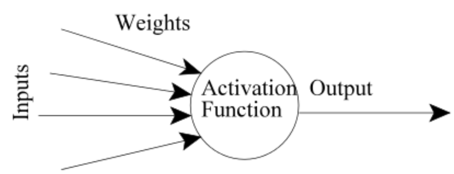

Figure 1: Artificial Neural Network

2.1.2 Layers

Typical Neural Net would have 3 types of layer that input layer, output layer and hidden layer. Each layer consists of one or more nodes and would pass information by flow to next layers. Particularly, input layer is layer that pre-process original data set that text data (DNN), Sequence Set (RNN) and images pixel data (CNN) etc into input vector. The hidden layers will work as assistant layers that assign different weight and neurons that help to construct the model. In the early time neural network typically has only 1 hidden layer [16] but recently with the development of deep learning approach, multi-layers is introduced into neural net that more than 1 hidden layer could have outstanding performance. Output layer is typically constructed with neurons that are label vectors. In training process output layers will produce the output labels and try to match the training labels for back-propagation that I will introduce in next section and after model is trained output layer would directly give the output based on given input data. Figure 2 demonstrates a neural network with 3 hidden layers and each hidden layers constructed by 9 neurons.

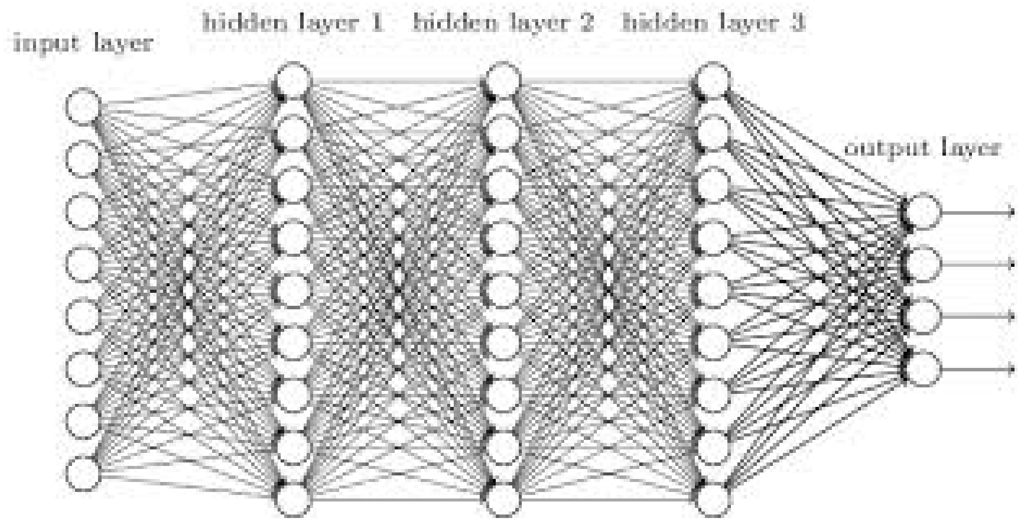

Figure 2: Artificial Neural Network [11]

2.2 Algorithm

The algorithm of neural network should ideally has 2 main parts that forward activation and backward propagation [25] [16] [8]. In this section, I would explain the mathematical definition of each algorithm and the walk through of the implementation.

2.2.1 Forward Activation

Considering each neuron is a perception, a forward activation function would be implemented in the neurons enable them to access, process and pass the information from previous neuron to next one. This stage is defined as Forward Activation [4]. During the activation process, An activation function would be implemented that take sum of product of weight w_i and input x_i

that obtained from all neurons in previous layers, processed by formula 1.

$$f(x) = f(\sum w_i x_i) \tag{1}$$

Typically, there are 2 most popular forward activation formulas that are mostly implied [8] that formula 2 Sigmoid and formula 3 Tanh.

$$f(x) = sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

$$f(x) = tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{3}$$

Figure 3 shows the graph of Sigmoid(left) and Tanh(right) as we can see clearly, the output range of Sigmoid range from 0 to 1 while Tanh ranges from -1 to 1. Normally for getting wider range of output information and getting better output closer to 0, Tanh would seems to be more popular than Sigmoid [11]

Figure 3: Sigmoid (left) and Tanh(right)

Overall, Forward Activation pass information from input layers to output layers though hidden layers. In that stage, raw information will be learned by the neural model perceptron based the weight and neuron architecture and eventually be output as a classification, clustered or regression result.

2.2.2 Backward Propagation

For updating the weights of each neuron, Back-propagation Algorithm will be introduced which is a common methodology to train the neural network.

The goal of this stage is to learn the entire weights in the network so that they have the actual output to be closer to the original target output, minimizing the total error for each output neuron and the network. The input of this algorithm would be the total standard error E_{total} we obtain in the forward activation stage. I would not explain the stage of mathematical derive and prove of this formula here, however after the the differential process the final formula will be formula 4 where o_j is output values of forward activation, y_j is the target labor of training data and η is learning rate.

$$w = w + \eta o_j(1 - o_j)(y_j - o_j)x_j \tag{4}$$

With the help of back-propagation, the weight can keep being updated until the iteration step reached. And we can get an optimized model by training it.

2.3 Resource

Recently, thanks to delicate work of researchers and software developers, we get number of open sources available with neural net and tool which can drive GPU to enhance to performance of training. In this study, I mainly used Tensorflow and Scikit Learning.

2.3.1 Tensorflow

Tensorflow [4] is open source library developed by Google aiming to use parallel GPU computation technique to construct, train and run the neural network. There are 6 maim parts of Tensorflow API body.

Tensor: The data structure of Tensorflow to store data includes constant value, sequence and random tensor used for different type of data set.

Variable: In-memory buffers containing tensors. Initial value defines the type and shape of the variable. Also, they must be explicitly initialized and can be saved to disk during and after training.

[4] `https://www.tensorflow.org/`

Operation: An operation serves as a node in a TensorFlow data structure takes zero or more Tensor objects as input, and produces zero or more Tensor objects as output, which can essentially serves as neuron or layers operation when constructing neural network.

Session: A class for running TensorFlow operations. Also in tensorflow API there is a particular session named Interactive Session which is a session for use in interactive contexts, such as a shell and Ipython notebook etc.

Place holder: A value that we'll input when we ask TensorFlow to run a computation.

Tensorboard: An API that can display training and testing result of Tensorflow by driving JS API.

2.3.2 Scikit-Learn

Scikit-Learn [5] is python machine learning library for researchers. A very decent advantage for Scikit-Learn is that it provides the simple interface for users for some machine learning algorithm including Support Vector Machine, Naive Bayes and so on. In this study, I mainly used Scikit Learn for 2 purposes that for one, use other machine algorithms API to evaluate with neural network in Scikit Learn. For another, take Scikit Learn as interface to drive Tensorflow using an open source library called Sickit Flow [6] which is library serves as interface using Facade Design Pattern to simplify the usage of Tensorflow.

3 Classification Problem

The first problem I resolve is supervised learning that classification problem. Supervised machine learning serves as an essential methodology in data mining industry. By inferring a function from labeled training data, analyzer can efficiently classify data and study important features of data set. Moreover,

[5] `http://scikit-learn.org/`

[6] `https://github.com/tensorflow/skflow`

it is probably the most widely used for of machine learning, that has been used to solve many interesting and common real-world problems [5]. The well-known problem in Supervised Learning is classification which is mapping from input set X to outputs Y where Y is a finite set of categorical variables. Specifically, given a set of record as training vectors assigned features and labels, we need to find a data model where class attributes work in the approximate function $Y = f(X)$ where is Y label of set. Particularly, I will concern on 2 different problems that text classification and image classification.

3.1 Text Classification

A very classical concern with classification is text classification, in this study, I will implement Deep Neural Network in Text Classification Problem.

3.1.1 UCI Letter Recognition Data

The letter recognition data set chosen for this task is created by David et al.. There are a large number of black-and-white rectangular pixels displaying as one of the 26 capital letters in the English alphabet. Character images are based on over 20 different fonts and each letter is randomly distorted to produce a file of 20,000 unique stimuli, where stimulus are almost equally distributed with mean 3.846% and std 0.113% as shown in Figure 4. Each unique stimulus stated above was converted into 16 primitive numerical attributes which were then scaled to a range of integer values from 0 to 15. Note that there are no missing attribute values.

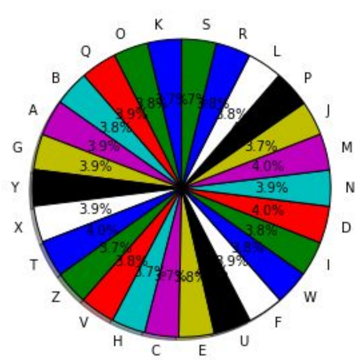

Figure 4: Class distribution in graphic model on Letter Recognition Dataset

3.1.2 Design Choice

As for choice building Deep neural network, in this experiment I set 3 hidden layers with 100 neurons associated each, and iteration to be 20000. For number of layers I have tried more than 3 layers but it turns out the accuracy of classification does not get significant enhancement. Also for the iteration steps, by plotting the loss values in figure 5, it clearly demonstrates that it loss rate has not reduce as clearly as previous before 20000 therefore to avoid the experiment to become meaninglessly expensive, the final decision is setting up 3 layers, iteration times to be 20000.

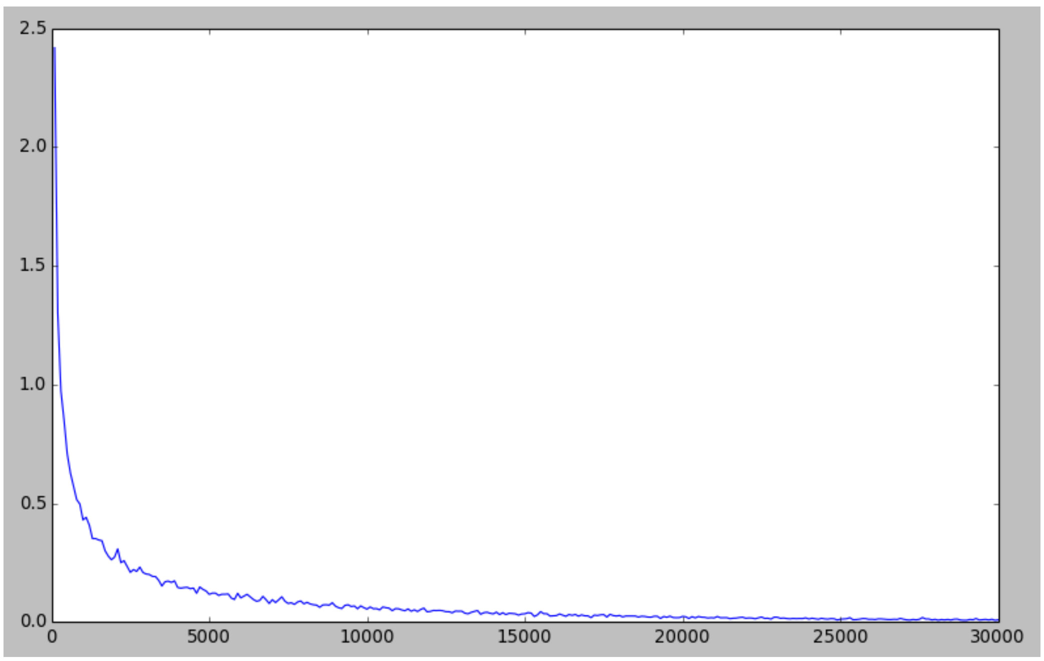

Figure 5: Loss rate of DNN among iteration time.

3.1.3 Result and Evaluation

The final accuracy result of the Deep Neural Network is 96.4% and time 54.69s for total training. Except neural network, researchers also develop a number of methodologies which are Naive bayes, Random Forest, Support Vector Machine as well as Adaboost. I selected the previous 5 methodologies which can relatively have well-performance in most cases of text classification

[14] [5] to compare with neural network.

Classifier	**Accuracy**	**Precision**	**Recall**	**F_score**	**Time**
DNN	0.964	0.964	0.964	0.964	54.69s
SVM	0.975	0.975	0.975	0.975	126.32s
DT	0.872	0.88	0.879	0.879	4.26s
RF	0.961	0.96	0.96	0.96	34.32s
Bst.DT	0.972	0.972	0.972	0.972	196.23s

Table 1: Experiment results on classification approach in terms of accuracy, precision, recall, f_score and running time. These results were generated from 10 folds.

As table 1 shows, we see that Deep Neural Network has a relatively high result and small training time. Support Vector Machine scores highest result but with nearly twice training time than neural Network. Decision Tree makes the time to be smallest however for accuracy result it only achieve 87.2% which is way less than Deep neural network 96.4%. Therefore generally, Deep Neural Network is an optimized methodology for this classification task.

3.2 Image Classification

Besides the text classification, image recognition is also considered as essential task for classification. By the previous work done by researchers, Convolution Neural Network(CNN) could potentially have outstanding performance dealing with such problems [15] [16]. Therefore in this task I would implement the Convolution Neural Network to classify the image. There are 2 main aims in this task, for one, to get the performance that accuracy of training result. For another, it is interesting to see that how convolution neural network learn the features of images and do classification, specifically visualize the convolution layers to see how each layer percept the features of images.

3.2.1 MNIST Handwriting Digit

MNIST data set is a handwriting digit images data set created by Dr Yann LeCun for image recognition research study with a training set of 60,000 examples, and a test set of 10,000 example. Specifically each example is a 28 pixels by 28 pixels handwriting number image.

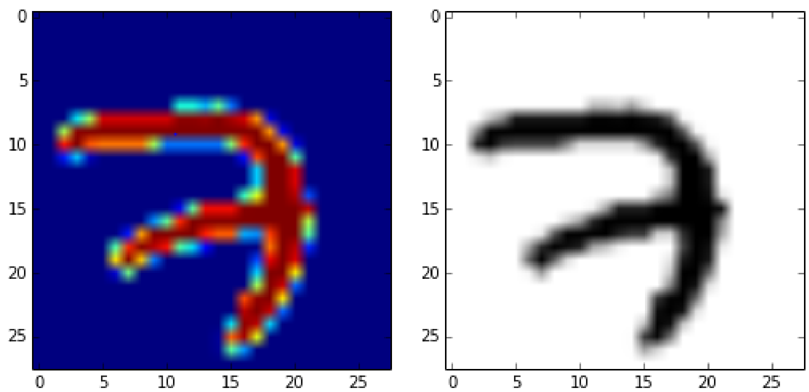

Figure 6: MNIST handwriting digit example

Figure 6 shows that the original and gray image of an example of a digit of an test data that training image with label 7.

3.2.2 Convolution Neural Network

By mathematical definition, convolution is a function derived from two given functions by integration that expresses how the shape of one is modified by the other[15]. In computer graphics theory, convolution processing normally serves as a linear filer to sharp the image that outline the features.

Convolution Neural Network is essentially nothing different from the deep neural network but take each neuron as feature map that convolution layer and use convolution filter as its weight.

Forward Activation of CNN is essentially to take convolution processing of each feature map to produce the activation map. In this stage the activation function like Sigmoid and Tanh would be applied to translate the each pixel of activation map. After that, the new feature map would be constructed by sum up all activation map of previous layer. In this stage, a sub-sampling technique(mostly max pooling [16] [15]) would be applied that for one to shrink the map and for another sharpen the features of image.

Backward Propagation of CNN is basically the same as DNN but instead of simply update scalar value weight, CNN will update the convolution kernal of each neuron by back-propagation. That the filters would learn and artificially know what features of image should be highlighted by training.

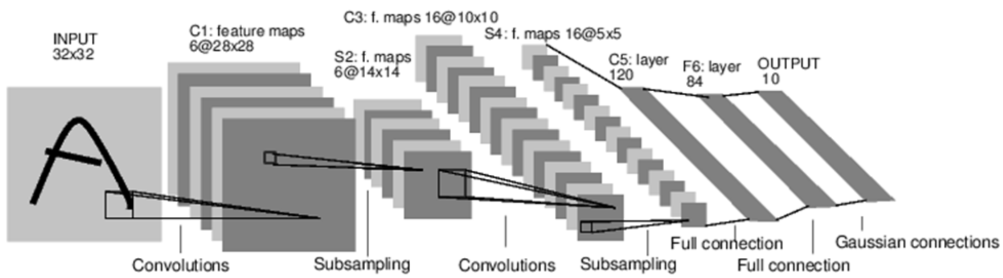

Figure 7: Architecture of Convolution Neural Network [15]

Figure 7 demonstrates that CNN, the feature maps in each convolution layer there is a convolution kernal to take convolution processing to generate activation map and by sub-sampling, the new feature map would be completed. For taking advantages of the 2D-matrix structure of an image or other 2D-formatted input such as a speech signal and so forth, the architecture of a CNN is designed achieved with local connections and tied weights followed by some form of down-sampling which enables translation invariant features.

In this experiment, I set up 3 convolution layers for CNN and set iteration number to be 10000.

3.2.3 Result and Evaluation

The Accuracy of CNN to this data set is 99.2% which is amazingly high. To further evaluate this result, I use softmax regression to train each example and compute a model. For instance, figure 8 is the model of number 7.

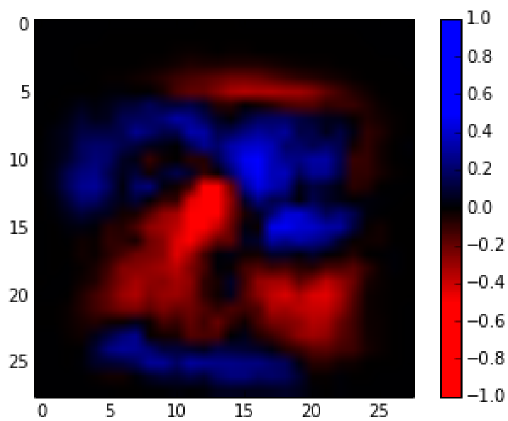

Figure 8: Softmax regression model for 7

The advantage of softmax regression is that it makes a model that is simple and fast to inference the input data to the output data, but its accuracy is obviously not as good as CNN. Except the high accuracy, CNN also has a more complicated learning processing. In this experiment, I input the figure 6 image in the trained model of Convolution Neural Net and visualize the each feature map to see how CNN learn this image data.

As shown in figure 9, each of 3 layers has shown feature of 7 that learn by CNN. There are 5 neurons I set up in first layers and as feature map (b) shows, the features learned by neuron is not much different from the original input images. However in second layers (c) 5 features map of is already being much different from original one but still in understanding of human visual perception of number 7 that is, we can still tell the feature map is demonstrating some 2D object that similar to number 7. Most interesting result is in third layers (d) which all of those feature maps are already far beyond our understanding which is eventual learning result of neural network, that how this neural net understand and define what this input images is. The result of this input image is number 7 that means the feature maps of third layer of this neural network is telling that it is how neural network interpret and study the image that can be defined as number 7.

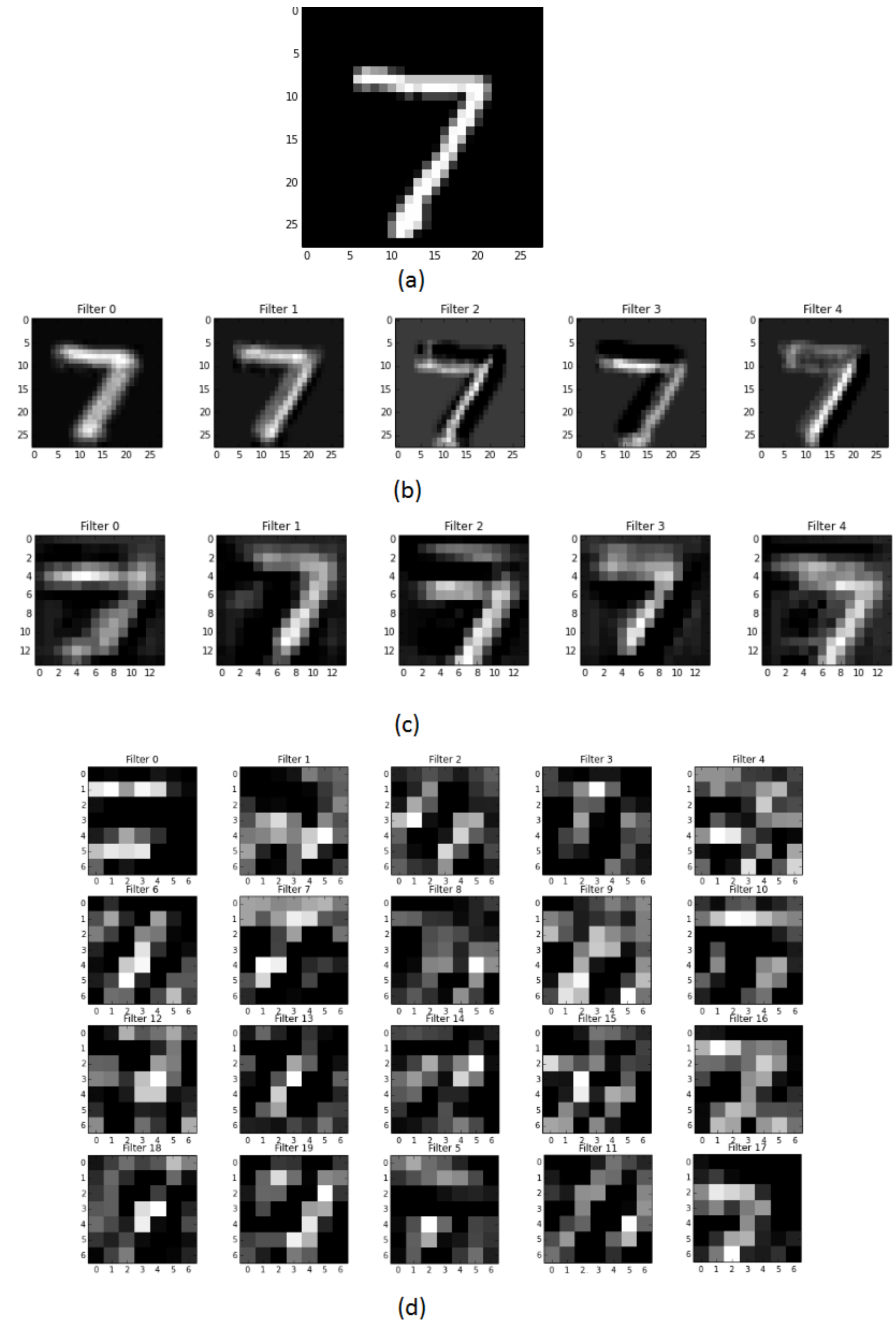

Figure 9: Visualize Convolution Layers of CNN Learning Test Image

4 Visual Analytic Problem

In this task, I proposed a system use neural network as back-end to visualize and solve the security problem. Specifically, neural network approach will be implemented in a visual analytic task that visual analytic of people movement data for security and management of a theme park. Different from previous task, this problem has been solved by the combine methodologies we proposed of neural network which is my own contribution to the problem. There are 2 types of neural networks would be applied for this task that for one, Self-Organizing Map(SOM)[13] and for another, Recurrent Neural Network(RNN)[18] to solve 2 main problems of safety issue based on movement trajectory data set given. The 2 problems are firstly the crime detection that the abnormal visiting pattern visualization and secondly the trajectory flow prediction that predict which attraction that visitors would likely to visit based on their previous visiting types. The data set is obtained from IEEE Visual Analytics Science and Technology Challenge 2015.

4.1 IEEE VAST 2015 Challenge

In the challenge, data set was on movement of visitors in a modest-size amusement park, DinoFun World, sitting around 215 hectares with 81 different attractions that can be classified by 7 main categories with Rides, Show & Entertainment, Information & Assistant, shopping, Beer Gardens, Restrooms and Food, where Rides can be further distinguished by Thrill Rides Kiddie Rides and Rides for everyone. All attractions are numbered, named and connected by a visitor pathway throughout the park[26]. Each visitor was tracked by a mobile device that records his/her positions in real time and his/her behaviors and communications. For protecting the privacy of users, devices only recorded two behaviors of customers that movement and check-in [23]. This park hosts thousands of visitor every day. Especially, there were 3357, 6411, and 7569 visitors on 6, 7, and 8 June 2014 respectively as an event “Scott Jones Weekend” was held for celebrating the coming of local star Scott Jones. On Sunday, 8 June 2014, a crime happened in this park and rapidly solved by officials [23]. Therefore, in this paper, I use this crime event as a specific case study to show how neural nets are used to detect the crime group and prove our hypotheses in task 1 while in task 2 I would used the approach to predict and flow of movement that would used

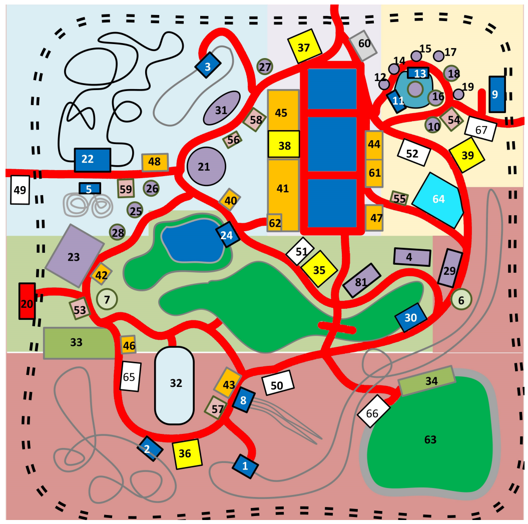

Figure 10: DinoFun Park Map

to enhanced the management system in the park security for big event. This data consists of two main parts: 25 million individual movement data (Mini-Challenge 1 (MC1)) and 4 million communication data (Mini-Challenge 2 (MC2))[9]. Specifically, movement data is in the format of (timestamp, person id, behavior type, position), while communication data is in the format of (Timestamp, call from, call to, location), and pre-processing of data is the first essential step in this study. The data are pre-processed in following ways:

- **Visiting Attraction Data** We pre-process the data from continuous 2D plane that geometrical position to be discrete attraction visit data, that we assume this person stays in nearest attraction of his/her location and that way we replace his location to the attraction label.

- **Check-in Frequency Data** Each visitor is represented as an 82-dimension vector, and each dimension of a vector represents each attraction. The value of each dimension represents the check-in frequency of this visitor.

- **Time Spend Data** Similar as check-in frequency data, an 82-dimension vector for each visitor represents 82 attractions and the value of each dimension of the vector is the time that visitor spent (in seconds) during the day.

- **Attraction Data** Tree data structure is used to store attractions with their position information as branches rooted by categories they belong to.

- **Movement Data Index** Movement data are indexed in two ways: visitor and attraction. This indexing allows to easily query information when we analyse trajectory, visit pattern and popularity of attractions as well as other features.

- **Communication Data Index** Each communication of 2 visitors are indexed by the visitors and time which allows us to see the communication pattern of visitors throughout the time.

The Check-in Frequency Data and Time Spend Data are used to represent the visiting pattern of each visitor in one day, which essentially shows the preference of that visitor. Specifically, we analyze how visitors spent their time in all attractions, time spent on each attraction and check-in frequency for each attraction. The data are used to cluster visitors and find abnormal movement behaviors Tree data structure allows to label the categories and attraction where visitors are currently in given the position data of visitors. Movement data index is used to get information for our visitor-oriented and attraction-oriented study. We also investigate frequency of communication between visitors.

4.2 Abnormal Pattern Detection

This section would explain the first task that using Self-Organizing Map to cluster the visiting pattern which is check-in frequency data and time spend data I processed previously to cluster. In this way, the people with similar or the same behavior pattern would be grouped for visualization.

Classical machine learning clustering algorithms of K-means[28] [1] [27] and K-Nearest Neighborhoods[28] are widely used for clustering trajectory data. However, such clustering methods cannot efficiently render a useful result especially for irregular movement patterns such as the people movement in a park. This is mainly because that K needs to be defined in advance for either K-means or K-nearest Neighborhoods and find an optimal K value is a non-trivial work. Therefore such clustering approach for trajectory data analysis cannot show much meaningful results[23]. These clustering methods are also sensitive to noise especially for massive cluttered movement data. Different from previous work, I use unsupervised approach of Self-Organizing Map to cluster people movement in this paper. Furthermore, most of previous work tries to cluster the movement trajectory directly [23], [24] [3]. However, such clustering become very difficult and even impossible when clustering cluttered and irregular trajectories. In our work, we try to cluster moving people based on their preferences of locations[27]. Visitors are clustered by the time spent in certain locations and check-in preferences. This kind of clustering method helps to find people who have the high potential to be in the same groups as they have similar or even the same interests. Based on interests and preferences of clusters, we can further infer what kind of visitors they are.

In addition, cluster visualization is a crucial in analytic of movement data[2]. Most existing visual analytics approaches try to model the trajectory information as movement patterns [28] [2]. Since our movement data with irregular and cluttered trajectories make the entire trajectory pattern meaningless, instead of discovering trajectory flow as most of previous approaches[3], we analyze people movement pattern according to their preference clustering. We also use different approaches[3] [27] to visualize the visitor preference pattern. Besides, other data such as communication, time spent on a site, preference categories are also considered as features for clustering. In this project, in this project neural network SOM would be applied fort this task.

4.2.1 Self Organizing Map

Self-Organizing Map (SOM) or Self-Organizing Feature Map (SOFM) is an artificial neural network invented by Teuvo Kohonen in 1982 [13]. In the field of data visualization, it is widely used to cluster and visualize high dimensional and non-linear data. The detailed description of SOM can be summarized in four major stages which are initialization, competition, cooperation and adaptation.

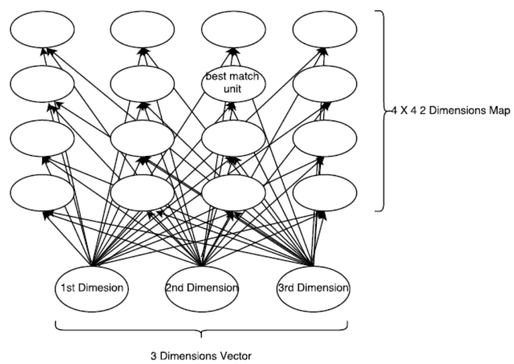

Figure 11: Construction of SOM Map

Initialization: Initialization creates a 2D-array map with nodes in the lattice and each node has an arbitrary value in its different dimensions. Fig 11 shows how SOM projects a three-dimensional vector onto a 4×4 2D lattice map. In our case we set the map size to 50×50

Competition: Competition uses Euclidean distance function as Equation 5 calculates the weight distance $Dist$ between every node W and input vector V in a 2D map. It then defines a node with minimum weight difference as Best-Matching Unit (BMU).

$$Dist = \sqrt{\sum_{i=0}^{i=n}(V_i - W_i)^2} \tag{5}$$

Cooperation: Cooperation uses radius decay function to locate the neighborhoods of BMU in each iteration. There are three sub-steps in this stage. The first step is to compute the distance between the node and the BMU with Equation 6.

$$BmuDist = \sqrt{(X_{bmu} - X_{node_n})^2 - (X_{bmu} - Y_{node_n})^2} \tag{6}$$

The second step computes a decay radius of neighborhood circle which takes the BMU as the center as shown in Equation 7. In Equation 7, $\sigma(t)$ is the decay radius of neighborhood, σ_0 is the initial radius of circle which is equal to the map width, t is the iteration time and λ is a constant number for learning rate which in our study set to be 0.05.

$$\sigma(t) = \sigma_0 e^{(-\frac{t}{\lambda})} \quad t = 1, 2, 3, ... \tag{7}$$

The last step is to compare every node to see whether they are inside their neighborhood circles. If *yes or* $BmuDist < \sigma(t)$, they are defined as neighbors of related BMU.

Adaptation: Adaptation trains all the nodes inside the neighborhood circle whose center is BMU using neighborhood function to update their weight W as in Equation 8.

$$W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t)) \tag{8}$$

where $\Theta(t)$ is used to consider neighborhood in the weight adaptation, $L(t)$ is the learning decay, for computing the learning rate in each training iteration. The factors that affect $L(t)$ are the current iteration time and the initial learning rate L_0. $L(t)$ is calculated with Equation 9.

$$L(t) = L_0 e^{(-\frac{t}{\lambda})} \quad t = 1, 2, 3, ... \tag{9}$$

The effect of neighborhood on the weight of nodes is the function of the distance between the current node and BMU, the radius of neighborhood circle, and the current training iteration time as shown in Equation 10.

$$\Theta(t) = e^{(-\frac{Dist^2}{2\sigma^2(t)})} \quad t = 1, 2, 3, ... \tag{10}$$

After iterative updating of all nodes, the map is self-organized and all nodes save their trained weights. The U-matrix of SOM is used to visualize the cluster map which shows the preference clusters of visitors. In the U-matrix visualization, the outstanding clusters are surrounded by very different and obvious colors which can be easily detected by users.

Different from regular neural network we know like DNN and CNN I introduced before, SOM is used to cluster the unlabeled data rather than doing classification. However for the **Adaption** that Self-organizing Map essentially uses the the same way as Back Propagation while **Cooperation** is similar as Forward Activation, we can regard this neural network simply as Neural Network plus a 2D rendering Map. Use self-Organizing Map we can see the people with similar or even the same visiting pattern would be clustered together therefore we can decode those visiting information to find out their visiting types that helps to detect the abnormal pattern.

4.2.2 Crime Pattern Visualization

As mentioned, in the visiting data of DinoFun World, the local soccer star Scott Jones had a celebration called "Scott Jones Show" at DinoFun World from 6 to 8 June 2014 over the weekend. All his personal honors including an Olympic medal were occurred at Creighton Pavilion (Attraction 32). There was a crime group vandalized the exhibiting, breaking into the place and did crime things on Sunday, 8 June 2014 [27]. Using Self-Organizing Map we would be able to detect this crime.

The visiting pattern of a crime group was usually different from the regular visitors [19]. Regular visitors normally have random distribution on the time spend and check-in in attractions in the park, while crime person may spend particularly longer time and higher frequency of check-in in certain places which is mostly the attraction that crime occurred. The trajectories of crime groups also may be different from normal visitors, for instance, when the show in Creighton Pavilion is temporarily closed and ready for the next show, normal visitors probably leave the place travelling to another place while crime people more likely still stay there during that time as it is good time to commit crime as no many visitors around there. From these assumptions, our visual analytics firstly conducts the clustering of data in the morning on Sunday. We then visualize the visiting pattern of outstanding clusters, and figure out their trajectory and communication behavior to detect crime groups.

In figure 12, visualization of SOM Map shows six distinct clusters. By further analyzing these six clusters, we found that among six clusters, two of them have the highest probability to be relevant with each other. We name these

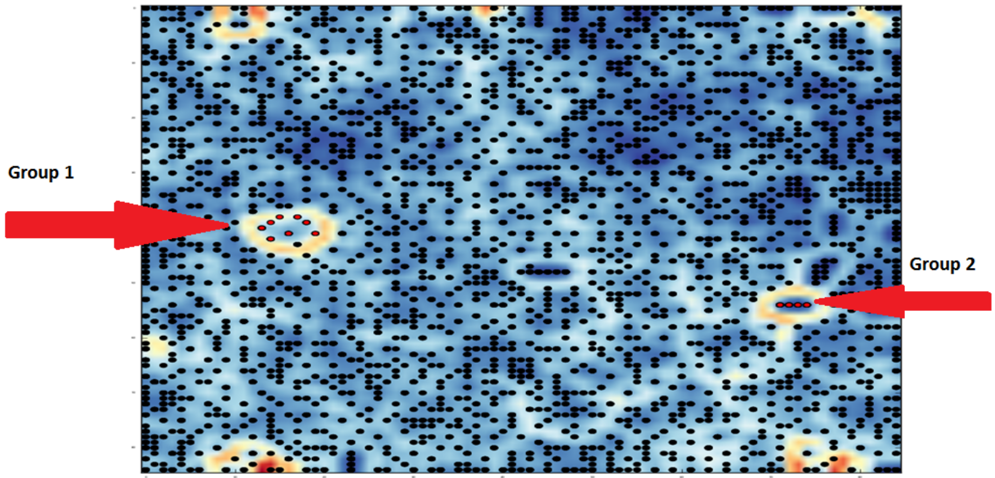

Figure 12: Clustering Result of SOM in Sunday

clusters as group 1 and group 2. By visualizing visiting patterns of these two groups as shown in In figure 13 we found that group 1 spent the most of their morning at two attractions of Attraction 53 Smoky Wood BBQ and Attraction 32 Creighton Pavilion, while group 2 spent almost the whole morning at Attraction 32 Creighton Pavilion. As both of groups spent quite a lot of time at the Attraction 32, we suspect these two groups committed a crime in Attraction 32 because of longer time spent at that attraction. By analyzing their movement in spatial-time cube visualization as shown in Figure 14, we found that from 9:30 am to 11:30 am, group 1 did check-in at Attraction 32 and stayed there until 11:00am, and then moved to Attraction 53. While group 2 spent all the time from about 9:30 am to 11:30 am at Attraction 32 without any further movement. In their communication visualization as shown in Figure 15, we found that the communication frequency of group 2 was much higher during the period from 10:00 am to 11:00 am than other time period. Therefore, we can assume that the time period from 10:00am to 11:00am was highly possibly the crime time. Based on these observations, we can conclude that group 2 committed crime while we still cannot exclude the suspects of group 1. We further visualize the communication pattern between group 1 and group 2 as shown in Figure 16, the result shows that group 1 and group 2 kept in touch during the whole day of 8 June 2014, which means that group 1 and group 2 were not independent groups and

Figure 13: Visiting Types of group 1 and group 2

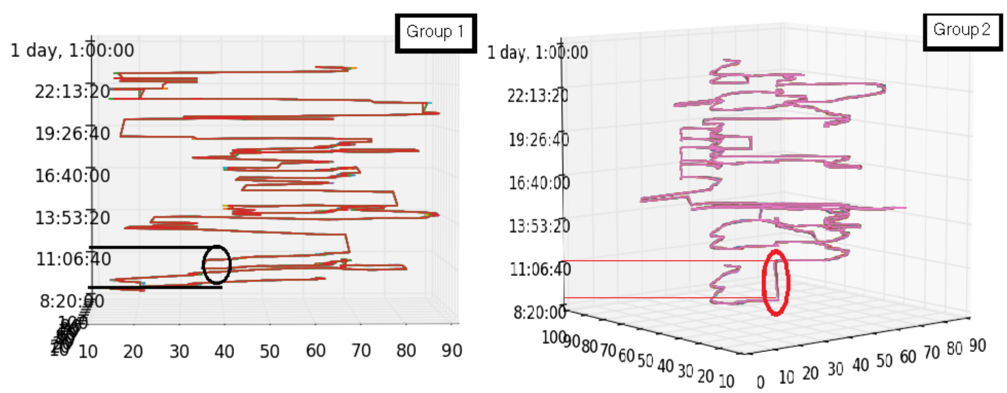

Figure 14: Trajectory Pattern of group 1 and group 2

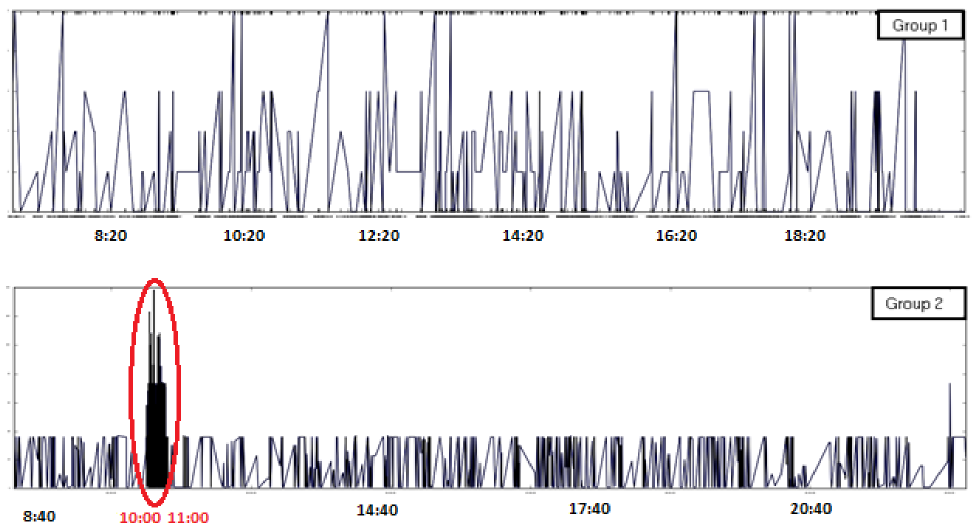

Figure 15: Communication of group 1 and group 2

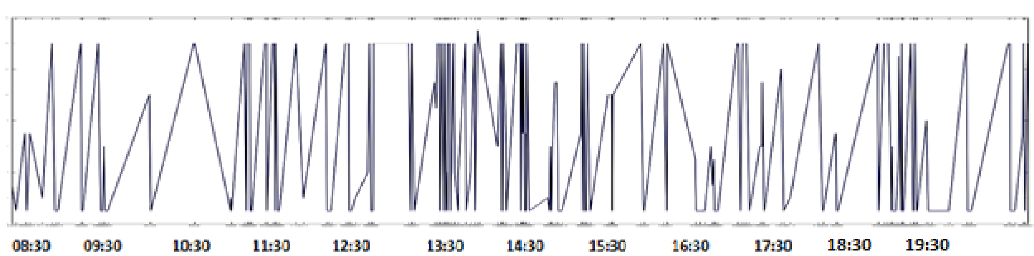

Figure 16: Communication between group 1 and group 2

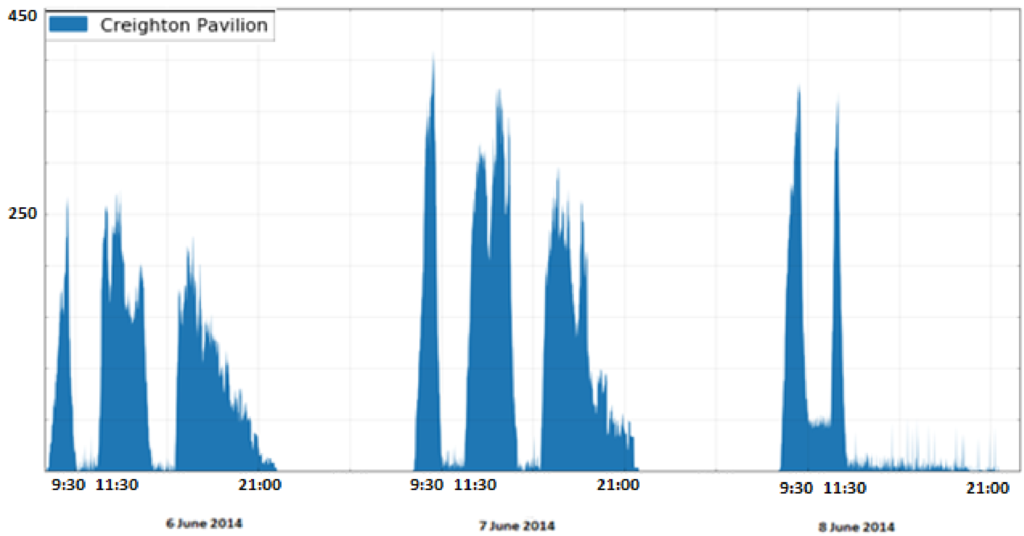

Figure 17: Visitor attendance at Attraction 32 Creighton Pavilion

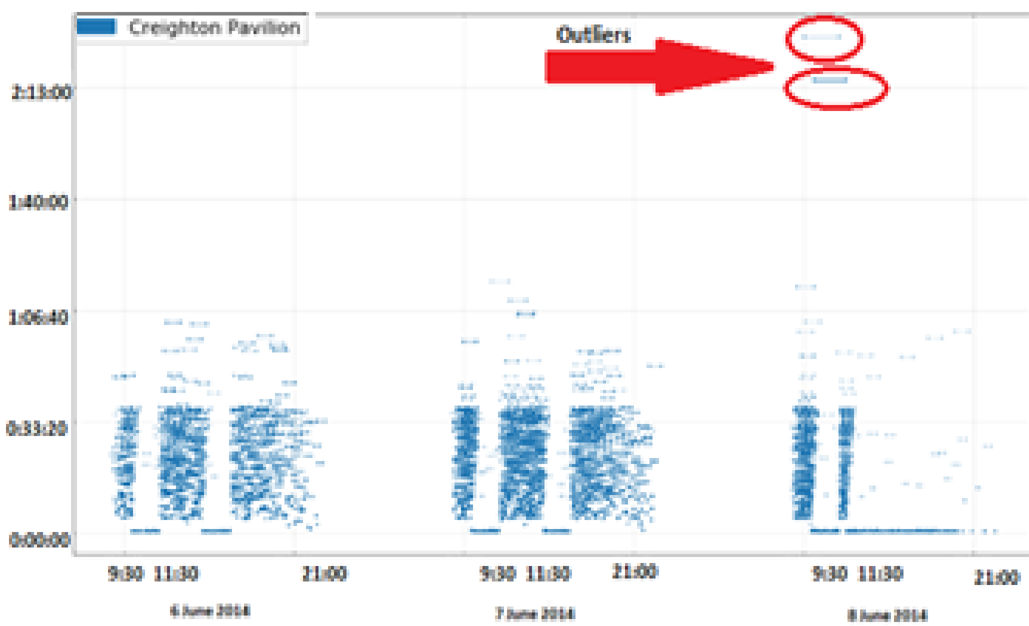

Figure 18: Visitor Time Spend at Attraction 32 Creighton Pavilion

they knew each other. Therefore, our conclusion is that group 2 committed crime at Attraction 32 while group 1 guarded Attraction 32 and Attraction 53 to help group 2 to commit the crime from about 10:00am to 11:00am.

Visiting data at Attraction Creighton Pavilion is further analysed to confirm our conclusions. Figure 17 and 18 shows the visiting pattern at Attraction 32 Creighton Pavilion. As shown in Figure 17, we found that there are three large attendance periods on each of three days, which means that "Scott Jones Shows" took place during these three time periods on each day. We also found that from around 9:30am to 11:30am every day, there was a sharp decrease followed by a very small number of check-in of visitors and then increased again. From this pattern, we can infer that from 9:30am to 11:30am this attraction was shortly closed until the next shows ready after 11:30am. Normally during this time period visitors most likely left this attraction. However from the time spend on Attraction 32 as shown in Figure 18, we found that there were two groups of visitors staying in this place from 9:30am to 11:30am. The time they spent at Creighton Pavilion are clearly outliers as normal visitors spent less than one hour there but they spent more than two hours in this place. Therefore, we can assume that these visitors are related to the crime.

In summary, based on these visual analytics as shown in this section, the whole crime story can be described as follows. "Scott Jones Shows" took place three times every day from Friday to Sunday. After the first show ending at about 9:30am, the attraction was closed until 11:30am. During this time period on Sunday, the crime people were divided into two groups: the first group mainly acted as assistance group and the second group committed the actual crime in Creighton Pavilion. The first group travelled around Attraction 32 to guard the crime. At around 10:00am, the second group started working on breaking into exhibition and stolen. Therefore the communication in the second group was increased a lot. Two groups also kept in touch frequently during the whole crime committing period.

4.3 Trajectory Flow Prediction

This section would go through how I use nerual network to predcit the trajectory flow of customers. In the visual analytic traffic flow prediction serves as a fundamental problem we consider for management of attraction based on customers movement. For specific instance, if we know in next hour, majority customers would be likely to move to certain attraction, more security guards can be issued there to avoid safety problem and assist the emergency cases.

To accurate and timely predict the flow, a lot of methodologies have been developed [22] [17] [12]. Different from the problems addressed by majority popular techniques, this project requires to solve the multi-trajectory problem that we do not just predict the flow of individual trajectory, instead, this issue covers all movements of vistors in the same time stamp that is we try to predict and visualize in certain time that how all the customer moves. This complicated problem is essentially combine number of individual predictions therefore it is trivial for multi-trajectroy problem assuming we address the single traffic flow prediction issue. In this case study, we take individual traveling pattern as sequence data similar as the way we perform natural language processing. Specifically, as mentioned, we have 82 different attractions, therefore, like what we do in previous task, we pre-process the trajectory from continues geometrical position using x, y 2D coordinators into discrete label data by given attraction labels. For instance a vector of an individual traveling would be (2,3,41,5) which means this person visit attraction 2, 3, 41 and 5 by observed and we need to predict what is the next attraction this person would be most likely to attend. Consequently, in particular case, we ignore the time stamp, and take the data as pure 1-dimensional sequence vector for training and take the predict result which is the attraction label as the vector label.

Due to the stochastic and non-linear nature of sequence data set, the researchers have paid much attention to the classical machine classification methodologies. Comparing with several methodologies, K Nearest Neighborhood, Random Forest and Support Vector Machine are all considered to be good ways for resolving such problem [12]. In this case study, I will use a dynamic neural network that Recurrent Neural Network and compare those methods to have evaluation.

4.3.1 Recurrent Neural Network

Recurrent Neural Work(RNN) is known as a popular network for training sequence data, it provides a dynamic training framework for neural nets [12] [7]. The feature of RNN is that the network contains feed-back connection, therefore activation can flow round in a loop which makes it able to do the temporal processing and learn sequences.

Architecture of RNN can varies different ways but they basically form in the Deep Neural Network structure that input layer, hidden layers and output layer which in hidden layers there are loops that can achieve dynamic updated and training based on time stamp t_i. Figure 19 shows the static model of RNN while Figure 20 shows the dynamic model of RNN

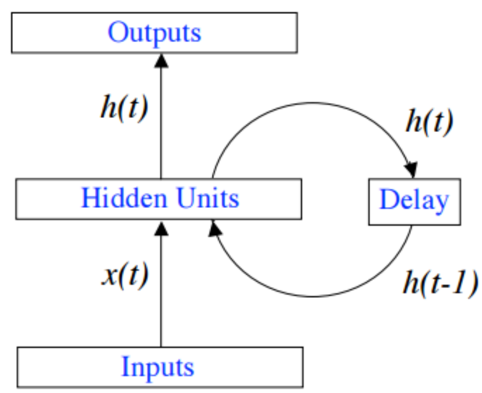

Figure 19: Architecture of RNN static model

Forward Activation of RNN is a mathematically dynamic feed-forward process that the result output from neural is depend on the sequence of data ordering with t_0 to t_i. Hence the forward function of RNN would be

$$h_t = \theta\phi(h_{t-1}) + \theta_x x_t \tag{11}$$

$$y_t = \theta_y \phi(h_t) \tag{12}$$

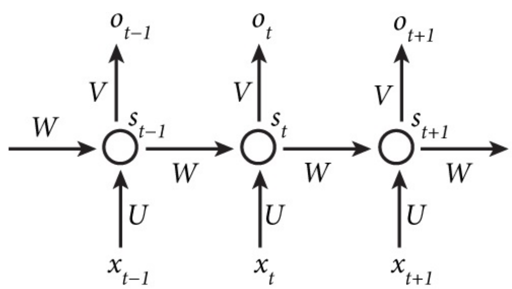

Figure 20: Architecture of RNN dynamic model

Where θ represent for the weights associated and ϕ is the forward function like Sigmoid etc.

Back-Propagation in RNN takes the time with essential parameter and we called it **Back-Propagation through time**, the formula of it can be presented as for figure 13 and 14.

$$E_{total}(t_0, t_1) = \sigma E_{sse/ce}(t) \tag{13}$$

$$\Delta w_{ij} = -\eta\sigma - \frac{\delta E_{sse/ce}(t)}{\delta w_{ij}} \tag{14}$$

Long Short-Term Memory (LSTM) [10] is a technique to solve the **vanishing gradient problem** [9] which occurs as time steps and each layer of architecture relate to each other through multiplication, derivatives would have potentially vanish. The work flow of LSTM is illustrated with Figure 21 which are setting up the output gating ,output squashing, memorizing and forgetting, input gating and squashing. LSTM would be useful to avoid such error that can be backpropagated through time and layers. They allow recurrent nets to continue to learn over many time steps (over 1000), keeping a more stable error, with opening a channel to link resources and effects remotely.

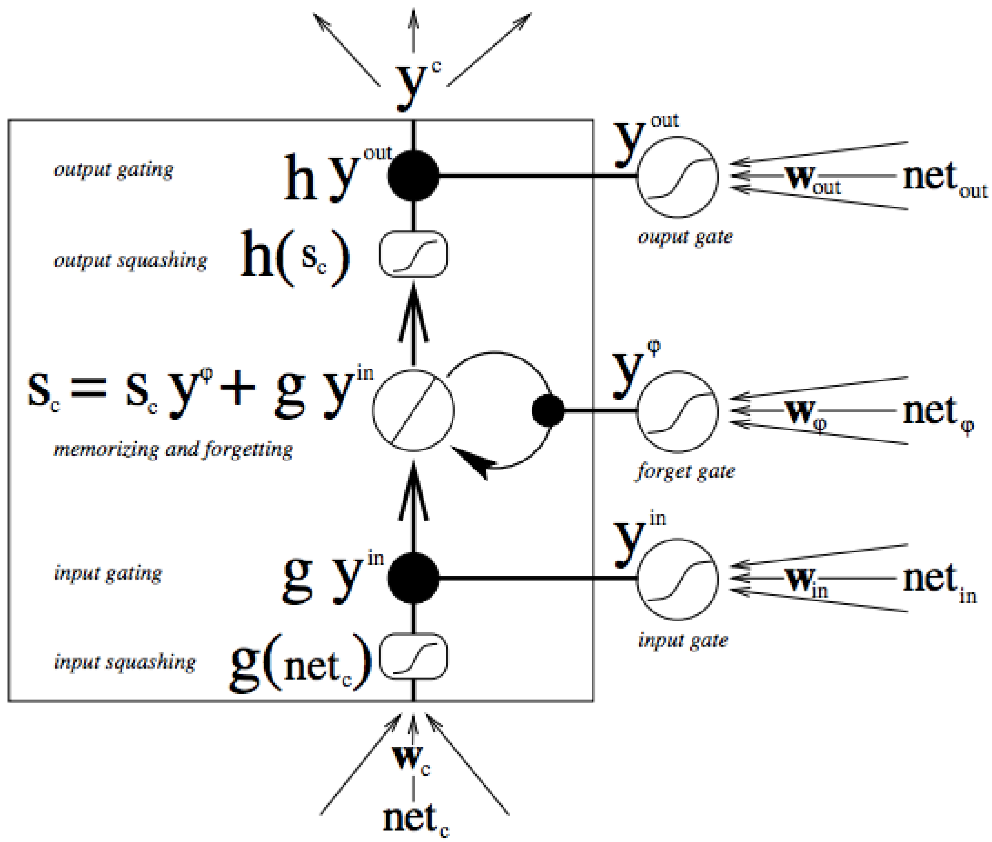

Figure 21: Long Short-Term Memory [10]

4.3.2 Prediction Result and Visualization

The result with using RNN to predict the work flow achieves the accuracy with 46% while the Non-ANN ways has the highest score 39% which is achieved by 10 Nearest Neighborhood. The total accuracy are presented by Figure 22 which we can see that RNN does shows an outstanding performance among the methodologies. Besides, we also have interest to see how RNN performs among the time in one day which illustrated by Figure 23 and obviously we can see that Recurrent Neural Network has gained almost all highest score among the time, and with the gross of training data(the time) the RNN increases significantly particularly after 20:00pm, the prediction therefore has a very high achievement.

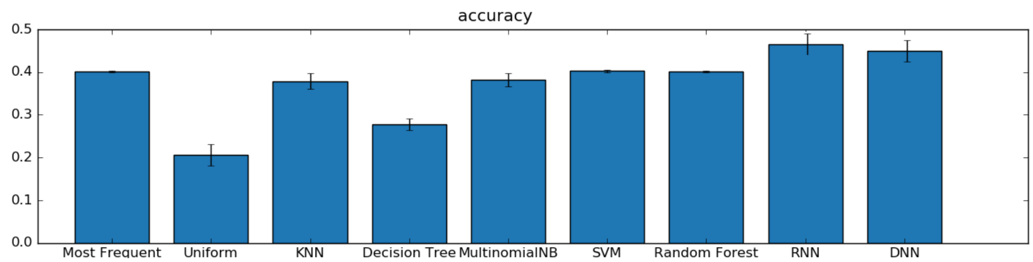

Figure 22: Total Score of Prediction

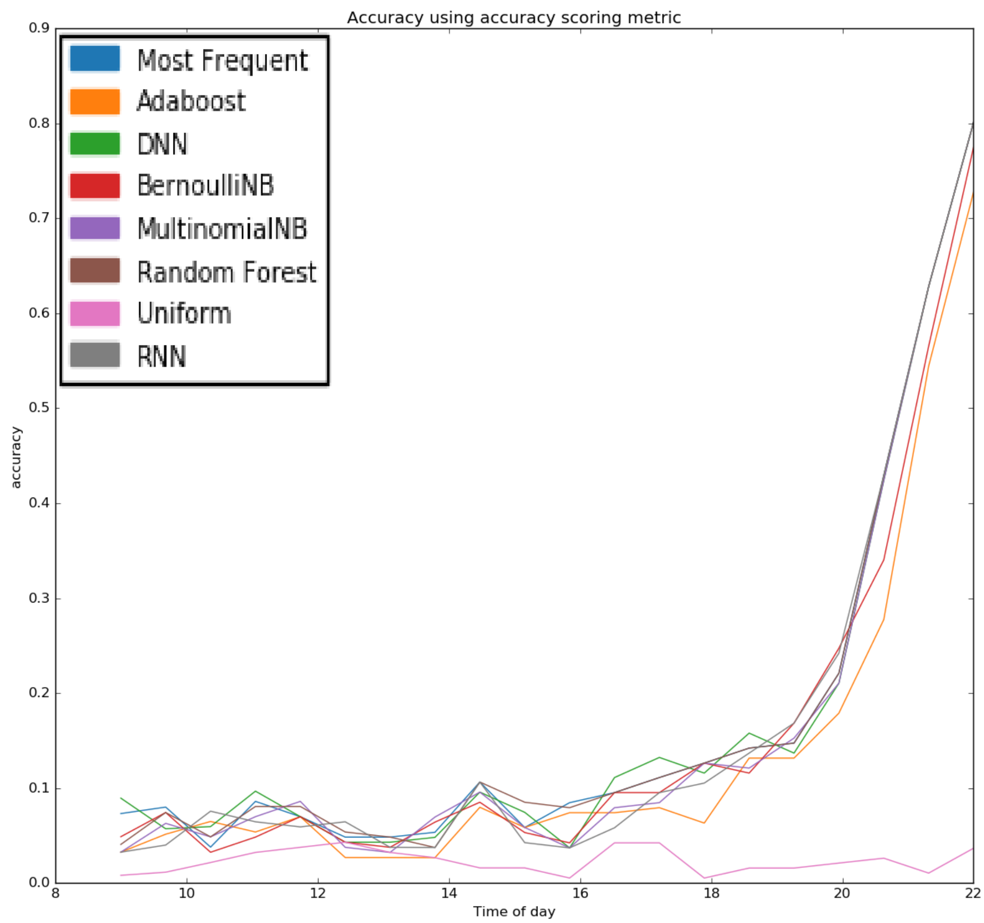

Figure 23: Score among Time of Prediction

To further discover the performance of this technique, I also visualize the flow by plotting them in the 2D park map. The experiment takes the movement for Friday 14:00pm data which is shown in Figure 24 and the thickness and color of flow arrows encode the information of the size of visitor group traveling from one attraction to another.

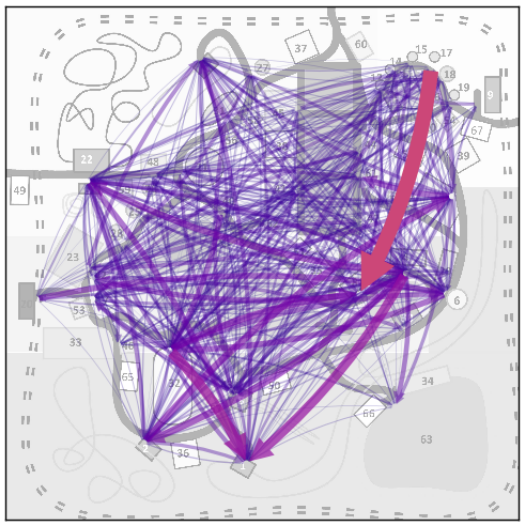

Figure 24: The Real Time Flow Prediction on Fri 14

From Figure 22 and Figure 23 we see that except the ANN ways (DNN and RNN) the best score is achieved by 10 Nearest Neighborhood, therefore I firstly visualize the prediction result of 10 NN and represent it in Figure 25. As we can see that the result of 10 nearest Neighborhood has the prediction which is obviously far beyond the actual movement flow therefore it is not considered as a good prediction.

Then we shows the prediction result which is obtained by Recurrent Neural Network demonstrated in Figure 26 that is apparently much better than what 10 Nearest Neighborhood gains and much closer to the real time flow that is more accurate. Hence, using RNN can be a proper resolution for this task that predict the traffic flow.

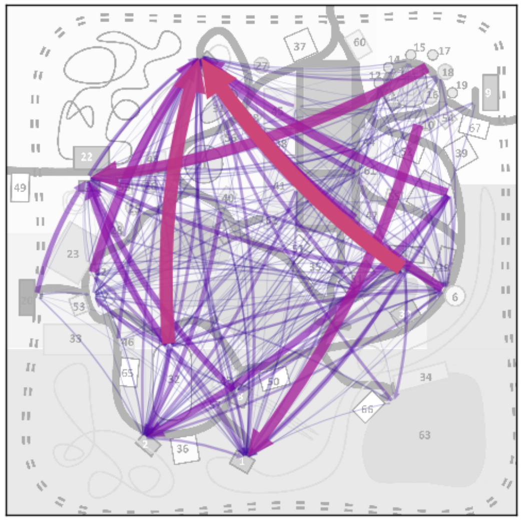

Figure 25: The 10 Nearest Neighborhood Prediction on Fri 14

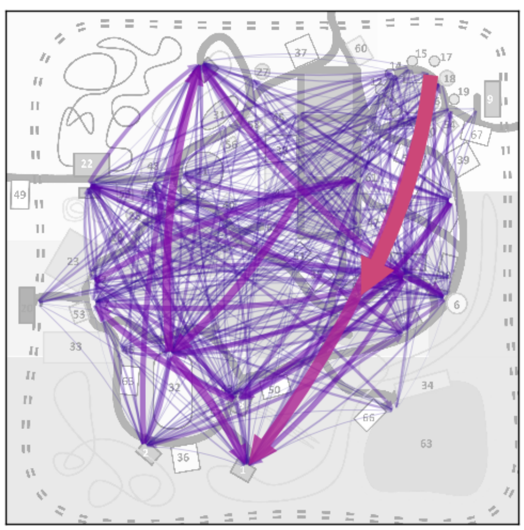

Figure 26: Recurrent Neural Network Prediction (3000 time training) on Fri 14

5 Conclusion and Future Work

This project mainly covers 4 different neural networks that DNN, CNN, SOM and RNN. These 4 neural nets are currently most popular ANN for research either classification problem and clustering problem in real industries especially for Natural language processing and Computer Vision. The first problem is more like the study as both of those 2 experiments has very easy solution that using neural network particularly for CNN in MNIST which is considered a very famous and typical experiment in computer vision [15] while the second problem is more to be an implementation of my knowledge on a real-world problem resolution. By using SOM and RNN as a back-end, the system shows the very good performance.

Throughout these 2 problems, it is very obvious to see the advantages of neural network that deep learning. With the help of advanced machine nowadays, neural network indeed has very sufficient work comparing other methods. However there are always 2 main problems associated with neural network that considered to be very essential. For one, what category of neural network that should be used for specific problem. Some neural net has been used as for common practice, for instance , Convolution Neural Net for image recognition as well as Recurrent Neural Net for Signal data. However it is also interesting that CNN can also be implemented for signal while RNN for computer vision task can also perform very well. Therefore choice for neural network is always a hot topic. For another, the architecture of neural net can also be varied even we choose one type. Design choice is also a good question to be studied and discussed, as different architecture can perform completely different. Future work of this study can be concerned one of particular neural network for a more specific problem and try more architectures of that type of neural net.

Acknowledgement

I would like to thank my supervisors Dr Jianlong Zhou in CISRO Data61 and Dr Xiuying Wang in USYD BMIT for their supervising and encouragement. Also I would like to give my appreciation to Dr Richard Xu in UTS Computer Vision Lab for providing the machine with GPU for neural network training.

Appendix

Other Experiment Results

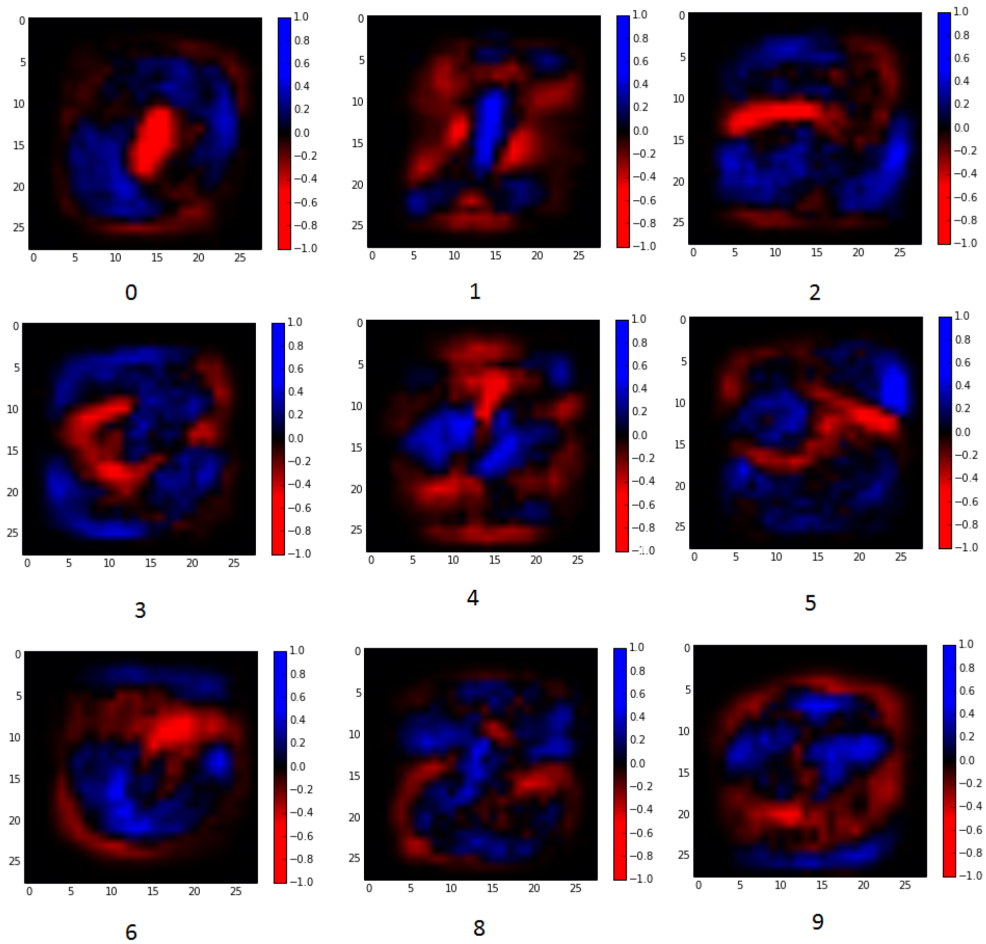

Figure 27: Softmax Regression Model for MNIST data 0 to 9

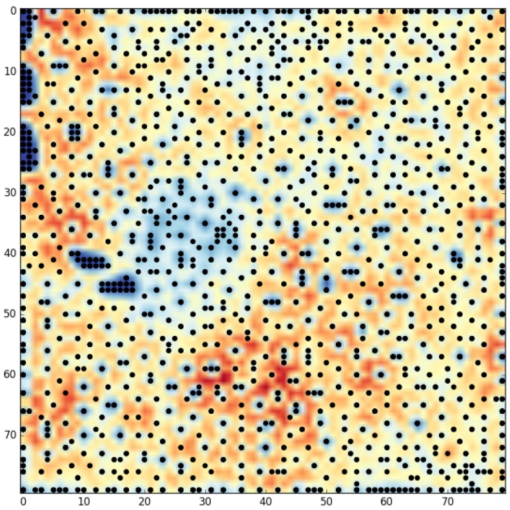

Figure 28: Self-Organizing Map Result for VAST Challenge 2015 Friday

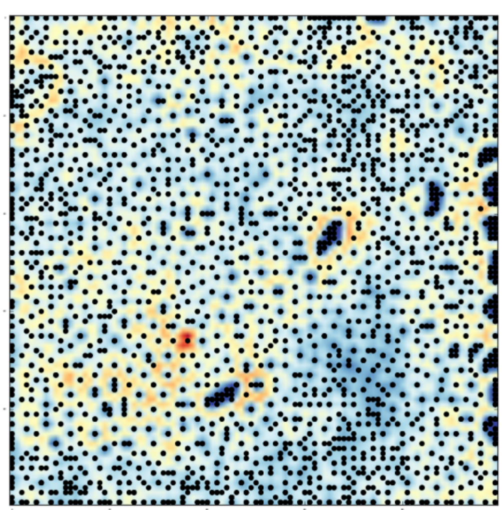

Figure 29: Self-Organizing Map Result for VAST Challenge 2015 Saturday

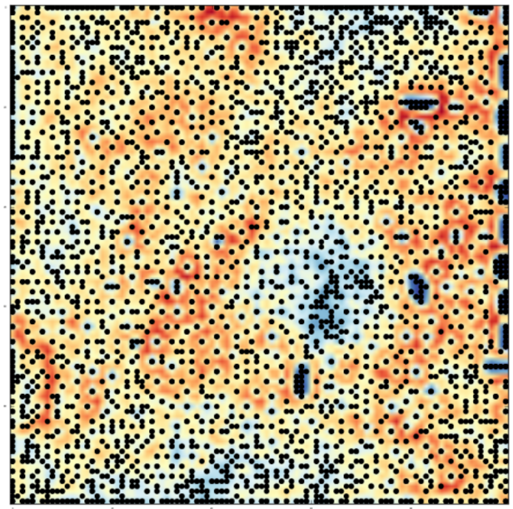

Figure 30: Self-Organizing Map Result for VAST Challenge 2015 Sunday

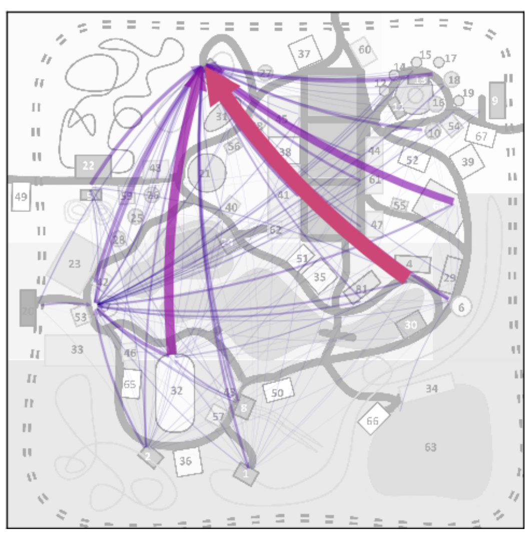

Figure 31: Random Forest Prediction Result 14:00 Fri

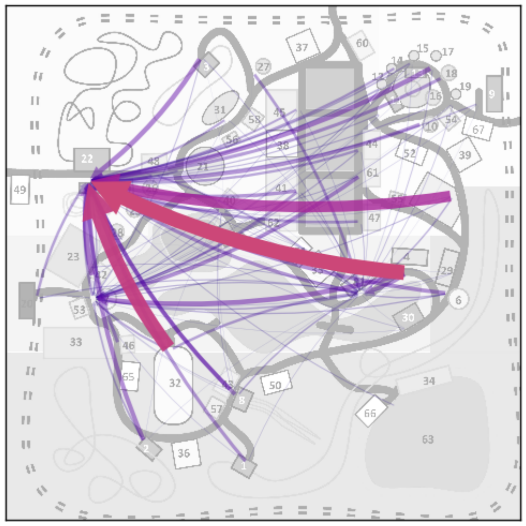

Figure 32: Ada-boost Prediction Result 14:00 Fri

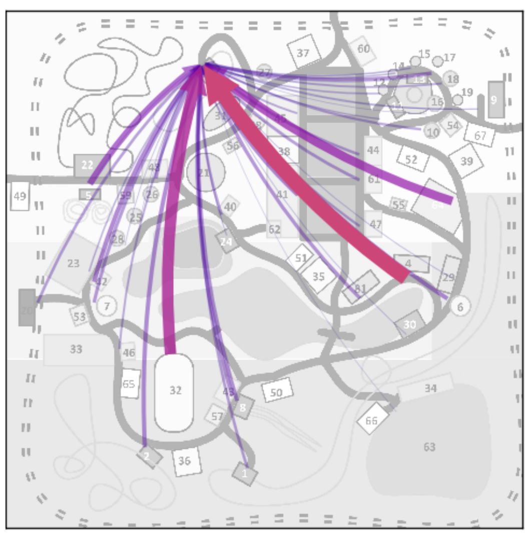

Figure 33: Most Frequency Prediction Result 14:00 Fri

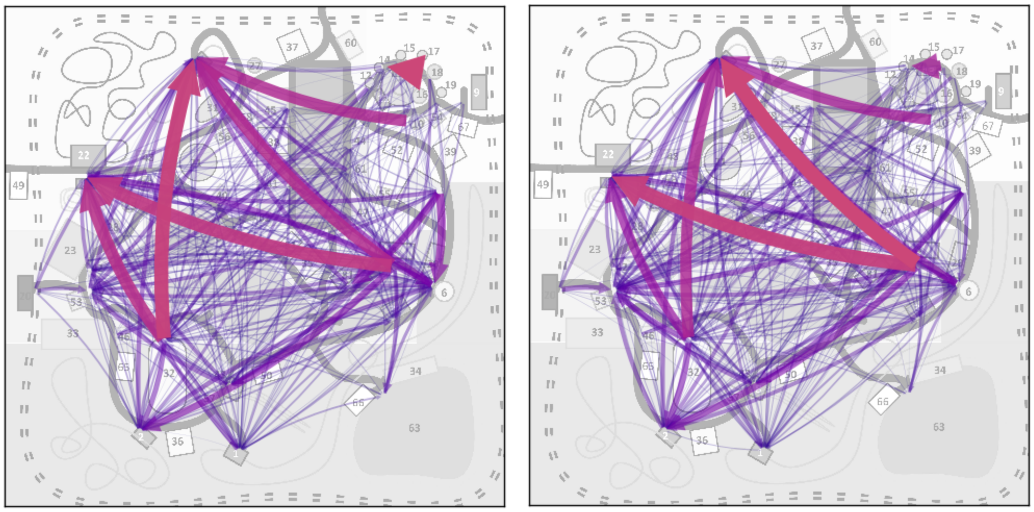

Figure 34: Naive Bayes Gaussian (Left) and Multi-normial (Right) Prediction Result 14:00 Fri

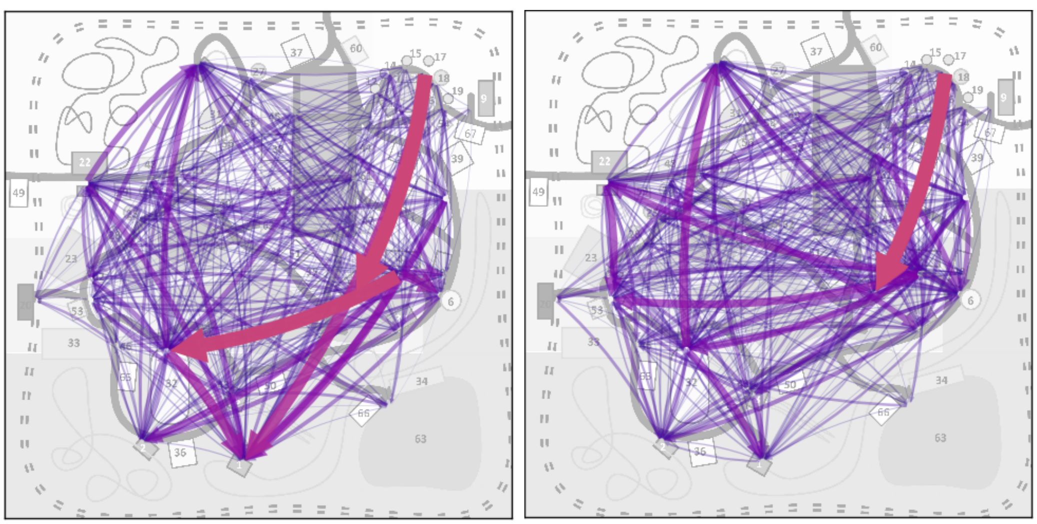

Figure 35: RNN 1000 times training (Left) and RNN 2000 times training (Right) Prediction Result 14:00 Fri

References

[1] Natalia Adrienko and Gennady Adrienko. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on visualization and computer graphics*, 17(2):205–219, 2011.

[2] Gennady Andrienko, Natalia Andrienko, Ioannis Kopanakis, Arend Ligtenberg, and Stefan Wrobel. Visual analytics methods for movement data. In *Mobility, Data Mining and Privacy*, pages 375–410. Springer, 2008.

[3] Natalia Andrienko and Gennady Andrienko. Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, page 1473871612457601, 2012.

[4] Elia Biganzoli, Patrizia Boracchi, Luigi Mariani, and Ettore Marubini. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statistics in medicine*, 17(10):1169–1186, 1998.

[5] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[6] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[7] Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, 6(6):801–806, 1993.

[8] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. PWS publishing company Boston, 1996.

[9] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[12] Wenhao Huang, Guojie Song, Haikun Hong, and Kunqing Xie. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2191–2201, 2014.

[13] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[14] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[17] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2015.

[18] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[19] Shyam Varan Nath. Crime pattern detection using data mining. In *Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on*, pages 41–44. IEEE, 2006.

[20] Costas Neocleous and Christos Schizas. Artificial neural network learning: a comparative review. In *Hellenic Conference on Artificial Intelligence*, pages 300–313. Springer, 2002.

[21] Berkman Sahiner, Heang-Ping Chan, Nicholas Petrick, Datong Wei, Mark A Helvie, Dorit D Adler, and Mitchell M Goodsitt. Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. *IEEE transactions on Medical Imaging*, 15(5):598–610, 1996.

[22] Brian L Smith and Michael J Demetsky. Short-term traffic flow prediction: neural network approach. *Transportation Research Record*, (1453), 1994.

[23] Michael Steptoe, Robert Krueger, Yifan Zhang, Xing Liang, Rolando Garcia, Sagarika Kadambi, Wei Luo, Thomas Ertl, and Ross Maciejewski. Vast challenge 2015: Grand challenge-team vader/vis award for outstanding comprehensive submission. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*, pages 119–120. IEEE, 2015.

[24] Tatiana von Landesberger, Felix Brodkorb, Philipp Roskosch, Natalia Andrienko, Gennady Andrienko, and Andreas Kerren. Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE transactions on visualization and computer graphics*, 22(1):11–20, 2016.

[25] Sun-Chong Wang. Artificial neural network. In *Interdisciplinary Computing in Java Programming*, pages 81–100. Springer, 2003.

[26] Mark Whiting, Kristin Cook, Georges Grinstein, John Fallon, Kristen Liggett, Diane Staheli, and Jordan Crouser. Vast challenge 2015: Mayhem at dinofun world. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*, pages 113–118. IEEE, 2015.

[27] Jieqiong Zhao, Guizhen Wang, Junghoon Chae, Hanye Xu, Siqaio Chen, William Hatton, Sherry Towers, Mahesh Babu Gorantla, Benjamin Ahlbrand, Jiawei Zhang, et al. Parkanalyzer: Characterizing the movement patterns of visitors vast 2015 mini-challenge 1. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*, pages 179–180. IEEE, 2015.

[28] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):29, 2015.