Crime Location Type Classification and Prediction

Lin Liu

***Abstract*— Crime pattern analysis is related to public safety and helpful for police patrols. There are three important topics, which are crime type prediction, criminal behavior pattern analysis and hot-spot prediction, in the crime pattern analysis. The prediction of crime location types is the problem in this project. In the data preprocessing, the features, which are hour, day of a week, business hour, and business day are extracted from date in the raw data. The column of output is descriptive words in the raw data and transformed into 4 categories. The 4 categories are home, public open space, public building, and others; however, the category of others is dropped before the data are fed into prediction models because this category provide no information about the crime location. Random forest is applied for the prediction of crime location types. Two important parameters, the maximum depth of each tree and the number of trees in a forest, are optimized. Feature importance indicates the relationship between the output and these input features. Two deep neural networks are implemented for the prediction. One deep neural network is a fully connected network with no dense embedding layers. There are dense embedding layers in another deep neural network. The one-hot encoded inputs are transformed into dense inputs in the dense embeddings. Important parameters, such as activation function and dropout rate of deep neural network are also optimized. The deep neural network models with different number of layers, number of nodes, and shapes are investigated. Then, the optimized random forest and deep neural network are evaluated in terms of precision, recall, f1-score, and computation time. The performance of deep neural network with dense embedding is better than deep neural network without dense embedding for all the criteria. The deep neural network performs better than random forest regarding the f1-score of the whole prediction results; however, the computation time for random forest is less than deep neural network.**

I. Introduction

Safety is a highly concerned public topic, which is related to the crimes. Crime is an important problem in every city, especially in cities like Chicago. According to the crime data for Chicago in 2016, the total number of crimes reaches 264679, which means 725 crimes happen per day in Chicago approximately. Analysis of crime pattern in a city is beneficial for allocating patrol resources and residents improving their protection consciousness [1]. In order to provide information about crime locations and alert police where to notice when they are patrolling, the prediction of crime location types is conducted in this project.

Lin Liu is with the Electrical and Computer Engineering Department, Master of Engineering, University of Waterloo, 200 University Ave. West, Waterloo, ON, Canada (e-mail: l289liu@ uwaterloo.ca).

Some of the crimes are random and rarely happen in an area. It is hard to find latent patterns in the crime cases and predict future cases, which means the prediction could be not so accurate. However, it is great even if there is a minor help for decreasing the crimes since it may save or help some persons from danger.

The field of crime prediction is similar to statistical inference, which is involved in a set of assumptions [2]. However, incorrect assumptions could lead to incorrect conclusions and invalidate the inference such as incorrect assumptions in the Cox models [3]. Traditional statistical methods analyze data for the population, race, surroundings, and education level in the crime prevention, which is a time-consuming task [4]. For the prediction problem in this project, it is a difficult task to provide a number of correct assumptions because of the complex relationship in the data set. On the contrary, machine learning is very powerful in solving such problems.

Some similar problems are presented in some studies. One problem is predicting the type of crime with time and the district in San Francisco. This problem is a Kaggle competition in 2015. However, the projects for the competition are not available on Kaggle. A problem similar to the competition is the blue collar/white collar crime classification in [1]. This problem is tackled using Gradient Boosted trees and Support Vector Machines [1]. In Vaquero Barnadas's project, he used K-Nearest Neighbors, Parzen windows and Neural Networks to predict crime categories [2]. Another problem related to crime pattern is the criminal behavior analysis, which is useful for investigating crime cases [5]. Crime hotspot maps is a widely used method of predicting locations of crimes, which is helpful for allocating polices in a city [6,7].

In this project, the problem about predicting crime location types is addressed. The crime location types are related to the primary crime types, for example, the theft is more common in the residential areas than in the public buildings. The community areas are important features for predicting crime location types because some community areas might be commercial areas with only public places. Community areas, primary crime type, and information related to time are selected as input features for the predictions. And the crime location types are separated into categories. Thus, the prediction of crime location types is a classification problem. Two methods are applied in the project for predicting the crime location types, which are random forest and deep neural network.

The report includes six parts. Section 1 is the introduction for the idea and the motivation of the project. The papers related to the crime prediction and the idea of deep neural network are discussed in section 2. Section 3 is a brief introduction of the theory for random forest, deep neural network and deep neural network with dense embedding. In section 4, a brief introduction of data and the preprocessing processes for the raw data are presented. Then, the procedures of parameters tuning in the random forest and in the deep neural network with dense embedding are discussed. Section 5 shows the prediction and evaluation results of random forest, fully-connected deep neural network and deep neural network with dense embedding. The comparison for these algorithms is discussed in terms of the accuracy and computation time. Section 6 concludes the experiments, results, and comparison for the two algorithms.

II. Related Work

The prediction of crime location types is not presented in papers to our best knowledge. However, the most similar problem is predicting crime types and hotspots, and there are some papers related to these topics.

Chandrasekar et al. analyze the prediction of crime types and implement some machine learning algorithms for solving this problem [1]. The features of the data from Kaggle are extracted first, and combined with additional census data of Chicago. Naïve Bayes classifier and random forest classifier are applied for prediction of crime types in 39 classes. The accuracy is less than 30%. In order to simplify the classification, the crime types are separated into two categories. The first separation method is blue collar or white collar crimes. Another method is separated according to the violent or non-violent crimes. Random forest, gradient boosted decision trees, and support vector machines are applied for the prediction, and compared in terms of accuracy. Gradient boosted decision trees and support vector machines perform well for binary classification problems.

Barnadas discuss crime prediction in his degree thesis. The problem is also the crime type prediction in 39 classes [2]. K-Nearest Neighbours (KNN), Parzen windows and neural network are used in the prediction. The results are submitted to Kaggle for the evaluation. The accuracy of neural network is the highest among the three. However, the KNN and neural network performed similarly because neural network is more complex in design.

In [8], lqbal et al. present their work for prediction of crime types, which is similar to the previously discussed work. The crime types are classified into “Low”, “Medium”, and “High” according to the popularity of crime types. These data ,which contain information for the US, are from UCI. Two widely used methods, bayesian classifiers and decision tree are applied for the prediction. Results of the two algorithms are presented with confusion matrix. The results indicate that the precision, recall, and accuracy for decision tree are much higher than those for Naïve Bayesian.

Another work is predicting crime patterns of four violent crime categories in [9]. The research is based on a communities and crime unnormalized dataset and crime dataset of Mississippi state. Linear regression, additive regression and decision stump algorithms are used for the prediction of crime patterns. The evaluation results with different criteria are discussed, and the linear regression is the best of the three algorithms.

Criminal behavior pattern is another topic for analyzing crime with machine learning algorithms. It is useful for behavior analysis of latent criminals and targeting at specific criminals when a case is investigated. In 2007, Adderley compared data mining approaches with the analysis crime specialists [10]. Multi-layer perceptron (MLP) is used for classifying burglary crime by two offenders. The cross-validation results indicate the approach is suitable for the classification. Self Organising Map (SOM) is a tool for unsupervised learning in the paper for clustering crimes. The author demonstrate that the two learning algorithms are effective in crime analysis. Keyvanpour et al. find criminal patterns in police reports with intelligent learning tools [11]. Critical information in the police reports is extracted. Then the authors proposed a method with SOM neural network for clustering these extracted data. SOM possesses the ability to reduce dimensions of dataset. The clustering results are separated into different categories, and MLP is applied for the supervised classification part. Two neural networks are used in the paper for different purposes.

Another source for identifying latent crime is social media because many people are active on these social media, such as Twitter. Gerber discuss the approach for predicting crimes with Twitter in Chicago [12]. They compare the results for test data with their approach and those with kernel density estimation, and prove their approach with Twitter could be used in actual system.

Hot-spot maps for crimes are available online for people to check out the danger and crimes. Kianmehr et al. present the prediction of crime hot-spots [13]. Support vector machine (SVM) is applied for predicting locations and compared with a statistical model SAR and neural network. Two-class SVMs perform better in terms of parameters tuning and efficiency of computation. The results show that the SVMs are appropriate for hot-spot prediction. Mohler propose a model for predicting hot-spot maps of crimes related to gun [14]. The model is marked point process model for predicting violent crimes in Chicago. Chicago is separated into 420 cells, and the location prediction is much higher than random prediction.

Kang et al. develop a deep neural network for predicting crime occurrence in Chicago [4]. A few datasets are used in this paper for the prediction, including crime data, meteorological data, pictures of Chicago, incomes, and education data. The environmental context information, neighborhood appearance, is addressed in this paper for the prediction. A convolutional neural network and a deep neural

network are combined for the prediction. Before the data are fed into neural network, the relationship between the features and crime occurrences is analyzed with Kruskal-Wallis H test. Convolutional neural network is applied for extracting information from image of Chicago. In the proposed deep neural network model, the features are tackled in separated layers and then merged together. The authors compared the performance of their prediction model with SVM and KDE. The deep neural network model performs better in terms of SVM and KDE. Other evaluation results indicate the importance of data selection and information about environmental context.

A review in 2015 discuss the data mining methods for supporting decision making (SDM) for crime prevention [15]. According to the authors, the most frequently used algorithms are Bayesian, neural network and nearest neighbor. These are the popular methods before 2015. In this paper, one of these popular methods are applied, which is neural network. But the neural networks are combined with some popular concepts recently.

The idea of implement deep neural network for predicting crime location types is from the papers of Zhang et al., Google Inc., and Guo et al. [16-18]. The target of their work is advertisement click-through rate (CTR) estimation. In this estimation problem, there are a large number of sparse input features, such as the cities all over the world of the users. Some features in the crime location type prediction are similar to the features in ad CTR estimation problem. In Zhang et al.'s paper, they applied factorization machine based neural network (FNN) and sampling neural network (SNN) for these sparse features [16]. However, the input layers are required to be pretrained. In the paper of Google, a wide and deep learning model is proposed [17]. Embedding is used in the deep component of this model, which is useful for tackling sparse input features. However, the wide component is directly connected with the output layer, which is not appropriate in the crime location type prediction. The work of Guo et al. is combined factorization machine and dense embeddings [18]. For the CTR prediction, there is only output which is different in crime location types classification problem. Their problem is an unsupervised problem while it is a supervised problem in our project. Based on their work, dense embedding and hidden layers are identified as important components for the prediction in this project.

III. METHODOLOGY

In this project, two different machine learning algorithms are conducted to predict crime location types. A brief introduction of two methods, which are random forest and deep neural network, is presented in this section. We also introduce the evaluation methods for the performance of these two methods.

A. Random Forest

A random forest classifier consists of a number of tree classifiers. An individual decision tree suffered from overfitting when the maximum depth of the tree is deep. A random forest possesses great ability of generalization with many trees [19]. It is implemented in different classification problem [20-22].

The random forest, an ensemble classifier, can be expressed as a set of classifiers, which are $h(x|\theta_1)$, $h(x|\theta_2)$, …$h(x|\theta_n)$. $h(x|\theta)$ represents a decision tree. n is the number of trees selected from a model random vector, and each θ_n is a randomly selected parameter vector. The final output y is [1]:

$$y = \text{argmax}_{p\in\{h(x_1)..h(x_k)\}}\{\sum_{j=1}^{k}(I(h(x|\theta_j)=p))\} \quad (1)$$

Random features are selected with replacement of all the features as inputs for the trees in the forest. Then, these trees in random forest are built separately with CART method and are not pruned [19]. The final decision is obtained through voting by every single tree or averaging probabilistic prediction. In our experiments, averaging probabilistic prediction is the method for making the final decision.

B. Deep Neural Network

A deep neural network is a neural network with multiple hidden layers. It is successful applied into many areas, such as natural language processing (NLP) [23, 24] and ad CTR estimation [16-18].

In our experiments, two structures of deep neural network are applied for the prediction of crime location types.

The first structure is a fully connected neural network with a few hidden layers. These hidden layers can be expressed as [18]:

$$a^{(l+1)} = f(W^{(l)}a^{(l)} + b^{(l)}) \quad (2)$$

where l represents the layer number, and f represents the activation function of each units in this layer, which could be tanh function, sigmoid function, or rectified liner unit (relu). $a^{(l)}$ and $b^{(l)}$ are the activations and bias in the l-th layer. And $W^{(l)}$ is the weights between *l-th* and *(l+1)-th* layer.

The output layer is also expressed as (2), f is softmax function, which is a widely used function in the output layer of a neural network [25].

The second structure contains dense embedding layers and fully connected hidden layers. These hidden layers and output layer are the same with those of the first structure. Dense embedding layer is used for transform discrete sparse input features into continuous dense inputs [16]. These categorical features are normally one-hot encoded in large dimensions. Dense embedding is useful to decrease the dimensions of matrix of inputs. The structure of embedding is in Fig. 1 [18].

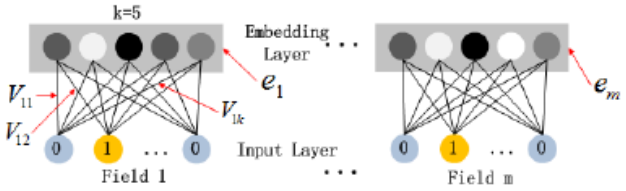

Figure 1. The structure of embedding layer [18].

For different length of input features, the length of embeddings is the same length *k*. In our deep neural network model, the *k* equals 4. The latent feature vectors are functioned as the weights in the neural network, which is similar to the previous work [18]. The output of the dense embedding is [18]:

$$a^{(0)} = [e_1, e_2, \ldots e_m], \tag{3}$$

where e_i represents the dense embedding of *i-th* field and m is the number of the fields. $a^{(0)}$ is served as part of inputs for hidden layers. In our experiments, the number of fields *m* equals 2. And another 4 non-sparse input features are directly fed into fully connected neural network.

C. Evaluation Methods

Precision, recall and f1-score are efficient strategies for the evaluation of learning algorithms. The precision is the percentages of the correct predictions for a specific label in all the predictions for this specific label. It is defined as [26]:

$$\text{Precision} = \frac{tp}{tp+fp} \tag{4}$$

where *tp* represents true positive and *fp* represents false positive for predicted labels.

The recall is the percentages of the correct predictions for a specific label in all true cases of this label, which can be expressed as [26]:

$$\text{Recall} = \frac{tp}{tp+fn} \tag{5}$$

where *tp* is true positive and *fn* is false negative.

The harmonic mean of precision and recall is f1-score. It represents the accuracy of a model, and it is defined as:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6}$$

These three criteria are applied in our experiments to evaluate the performance of random forest algorithm and deep neural network models.

IV. Experiments

Two different approaches, which are random forest algorithm and deep neural network, are explored in our experiments in order to predict crime location types. All the experiments are conducted on the Pycharm platform with Python 3.5.2 in my Dell (Inspiron 15 7000) computer with Windows 10 operation system.

A. Data Preprocessing

The dataset for crime information of Chicago is downloaded from Kaggle, which is a public website contain many free datasets. This dataset includes the crime data in Chicago from 2001 to 2017, and we select the data from 2012 to 2017 as our training and test data. Table 1 shows some sample rows of the dataset.

TABLE I. Sample Rows from Raw Dataset

ID	*Case Number*	*Date*	*Primary Crime Type*
0	HZ250406	05/03/2016 11:40:00 PM	BATTERY
1	HZ250409	05/03/2016 09:40:00 PM	BATTERY
2	HZ250503	05/03/2016 11:31:00 PM	PUBLIC PEACE VIOLATION

ID	*Location Description*	*Community Area*
0	APARTMENT	29.0
1	RESIDENCE	42.0
2	STREET	25.0

There are many features in this dataset, and we select a few features for predicting primary location type. The selected features are “ID”, “Case Number”, “Date”, “Primary Crime Type”, “Location Description”, “Community Area”. The records which contain null values of the selected features are dropped before preprocessing.

ID and Case Number: There are unique numbers to indicate a specific incident. ID is the identifier in the dataset and Case Number is the Chicago police department Records division number.

They are applied in the data preprocessing for identifying whether there are duplicate records.

Date, Primary Type, Community Area: Date is the time of the incident. The format of date is “month/ day / year hour:minute:second PMorAM”, for example, “01/01/2014 12:01:00 AM”. Primary crime type is the category of an incident, and the total types of incidents are 39 types. Primary crime type is presented in description words, such as “THEFT” and “SEX OFFENSE”. Community area indicates where the incident occurred. There are 77 community areas in Chicago. They are described using the numbers from 1 to 77 in the dataset.

The information in the date are extracted with Pandas in the Python. Hour in a day, day of a week, business hour, and business day are important features of the date. Hours in a day are separated into 24 categorical numbers. Seven numbers from 0 to 6 represent the different day of a week. 0 represents Monday, and 6 represents Sunday. Business hour and business day takes two values 0 and 1. 0 represents false while 1represents true. The description words for primary type are

transformed to categorical numbers through LabelEncoder in scikit-learn.

Hour in a day, day of a week, business hour, business day, primary crime type, and community area are all the input features for the prediction of primary location type. They are all transformed to numbers before fed into the training.

Location Description: These are brief descriptions of the location type for the incidents, which are not in a standard words and form. There are 142 different descriptions, and only one case for some descriptions. It is hard to build a model to predict the type of a crime in 142 categories. However, these primary location types can be separated into a few categories according to the features of the location types. Thus, we divide the locations into four categories, which are home, public open space, public building, and other location. We separate the 142 location types into these four with the key words in the descriptions. For the "other location" category, it is meaningless to predict whether a location type in this category because it provides no useful information to the police. We did not mean to add this category, some descriptions in the original data are "OTHER". The cases in this category is less than 10% of the whole dataset so that these data are simply dropped.

The location types (home, public open space, public building) are the output labels in the prediction, which are presented as numbers. The numbers 1, 2, and 3 represent home, public open space, and public building, respectively.

The crime data for training a learning model is the crime data from 2012 to 2016. The number of training dataset is 1350066 records after the duplicate incidents are dropped.

The crime data for the prediction and test a learning model is the crime data in 2017. There are totally 10551 records in the dataset after the duplicate cases are dropped.

B. Algorithms Implementation

Random forest: An ensemble API RandomForestClassifer is applied for the implementation of random forest algorithm. After loading the dataset, the datetime is used as the index of the dataset. We first separate the training data (2012-2016 crime data) and the test data (2017 crime data) according to the year in the index. Then the random forest is trained with different parameters, and the outputs of test data are predicted with the random forest. The comparison results of predicted and true labels are presented by using the confusion matrix in the Scikit-learn library [27]. The confusion matrix was plot with the matplotlib, which is a common tool for plotting on the Python platform.

The classification-report in Scikit-learn is applied to present the precision, recall and f1-score for all the labels and the whole results.

The "time" method is also applied in the program in order to compute the time required to build a model and predict the outputs.

Deep neural network: The neural network is based on the layers provided in the Keras. Two methods for building neural network models are used in our experiment.

The first method is Sequential model for building a neural network without dense embedding for sparse features. We added dense layers with different activation function and compiled the model in order to build a fully connected neural network.

The second method is Model, which is more complex than Sequential. It is applied to build a neural network with dense embeddings. Some features are sparse features, for which dense embedding are required, while some features are non-sparse features. The difficulty for building this model is that adding a layer using Sequential for all the inputs is not suitable because embedding layers for part of inputs are required. It is possible with Model by define a layer for merging different basic layers together. Another problem in Keras is that a flatten layer is required for connecting the dense embedding layer and fully connected hidden layer. The flatten layer transform 3D tensor to 2D tensor.

The two models with different hyperparameters are trained with the same dataset in the random forest training. And the prediction for the labels of the test data are conducted.

Confusion matrix, classification-report, matplotlib, and time are also applied for the evaluation of the neural network models.

C. Parameters Tuning for Random Forest

The number of trees and the depth of each tree are two important parameters in the random forest classifier. The performance of different random forests with different values of the two parameters is compared in order to select the most optimal values.

Fig. 2 presents the variation of accuracy for different maximum depth of each tree and 50 trees in the random forest classifier. With 50 trees in each random forest, the maximum accuracy obtained was 0.5898 when the depth of tree equals 15. The precision and recall for public building class are 0 with shallow trees.

The accuracy of the prediction with different number of trees in the random forest is presented in Fig. 3. The maximum accuracy of random forest is 0.5934 with 40 trees in the random forest and the depth of each tree is 15.

For the random forest algorithm, the importance of all the features is presented in fig. 4. The most relevant features are primary type and community area. The feature business day is almost irrelevant to the crime location types. These features can be selected according to the importance level.

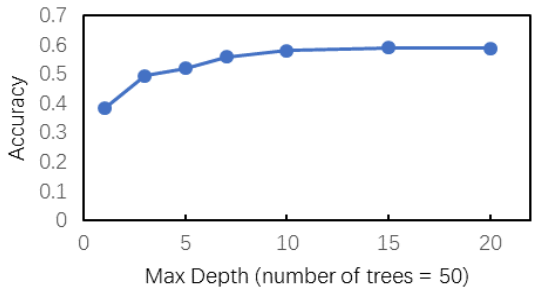

Figure 2. The variation of accuracy with different maximum depth of each tree. *(number of trees = 50)*

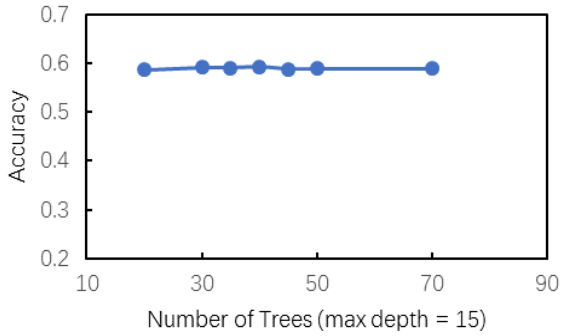

Figure 3. The variation of accuracy with different number of trees. *(max depth = 15)*

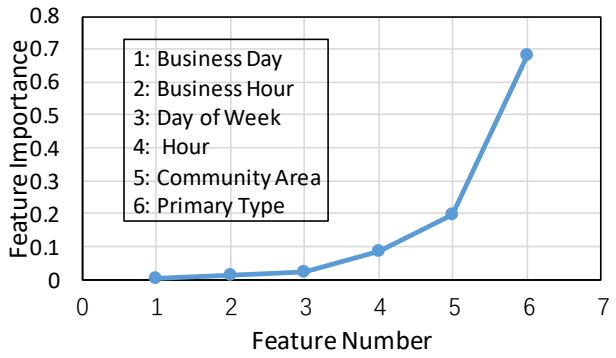

Figure 4. *Feature Importance*

D. Parameters Tuning for Deep Neural Network

A deep neural network involves many salient parameters and details. A number of details are discussed before presenting the optimized neural network model. And the neural network with dense embedding layers is the example in this part.

Stochastic gradient descent is the optimizer in our experiments. We apply early stopping to determine the number of epochs similar to [16]. The training will stop when the decrease of loss for validation data is less than 10^{-5}. The validation data are the 2017 crime data. The activation function in the output layer is softmax function. Fig. 5 is the neural network diagram with dense embeddings in the experiment. And we also conduct an experiment on neural network without dense embedding.

Three activation functions in the hidden layers are compared in terms of the accuracy of classification. In this experiment, there are 128 units in each layer and 2 hidden layers in the neural network. Table 2 presents the performance of neural networks with different activation function. The test data for accuracy evaluation are 2017 crime data. The performance of network with tanh function in the nodes is better than that of networks with relu or sigmoid function in regard to the accuracy and computation time.

The number of hidden layers and the number of nodes are two important parameters regarding the architecture of a neural network. The neural network is more complex with more layers and nodes so that the computation time increases with the number of layers and nodes.

We compared the accuracy and the computation time for different models with 2, 3, 4, and 5 hidden layers. Table 3 shows the performance of different number of hidden layers when 128 nodes in each layer and tanh function is the activation function in these nodes. The accuracy of the network with 4 hidden layers is the best among the four. The computation time of the network with 4 hidden layers is a little longer than that of the network with 3 hidden layers. When the structure is 5 hidden layers, the computation time increase sharply, which is more than twice of that in 4 hidden-layer structure.

The number of nodes in every layer can be in a large range. It is very time consuming to compare all these different models. In our experiment, we choose the optimal number for the nodes from 64, 128 and 256 in a 3 hidden-layer neural network. The accuracy is 0.5621, 0.5890, and 0.5801 for 64, 128, and 256 nodes, respectively. And the computation time of 256 nodes is much higher than 128 nodes, so 128 is the optimal number among the three.

Dropout become a popular method to prevent complex co-adaptations on the training data [28, 29]. The neural units are omitted randomly during training. Dropout rate determines the probability that a unit is dropped in the training process [30].

The dropout is applied in our neural network structure for preventing overfitting. The influence of dropout is tested and the results are presented in fig. 6. When the dropout rate is 0.9, all the prediction labels are label 2 so that the accuracy might not be suitable. The optimal dropout rate for the neural network is 0.5, while the performance of networks is similar when the dropout rates are in the range of 0.3 to 0.5. For some problem, the smaller dropout rate might be chosen since the

computation time is increasing with the increase of dropout rate. In our case, dropout rate of 0.5 is the optimal because there is a minor difference of computation time for dropout rate less than 0.5.

Four network shapes, i.e., constant, increasing, decreasing, and diamond, are also investigated. In our experiment, four shapes neural networks with three hidden layers are constant (128-128-128), increasing (64-128-256), decreasing (256-128-128), and diamond (64-256-64). Table 4 is the comparison of performance for the networks with these four shapes. It is obvious that constant shape is the best structure shape among the four regarding the accuracy and computation time. The result of accuracy is consistent with [18] and [31].

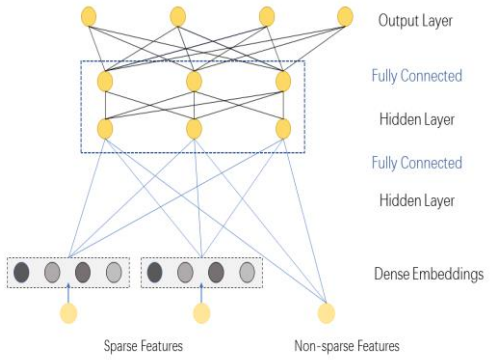

Figure 5. *Neural netwrok diagram with dense embedding*

TABLE II. PERFORMANCE WITH DIFFERENT ACTIVATION FUNCTIONS

Activation Function	Evaluation	
	Accuracy	*Computation Time(s)*
relu	0.5308	184
tanh	0.5777	143
sigmood	0.5766	420

TABLE III. PERFORMANCE WITH DIFFERENT NUMBER OF HIDDEN LAYERS

The Number of Hidden Layers	Evaluation	
	Accuracy	*Computation Time(s)*
2	0.5777	143
3	0.5890	156
4	0.5988	181
5	0.5890	429

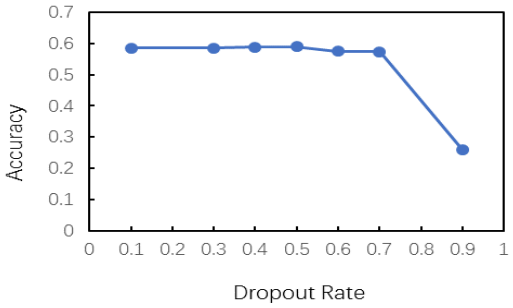

Figure 6. The variance of accuracy with dropout rate

TABLE IV. PERFORMANCE WITH DIFFERENT SHAPES

Shape	Evaluation	
	Accuracy	*Computation Time(s)*
constant	0.5890	156
increasing	0.5831	216
decreasing	0.5575	217
diamond	0.5707	187

V. RESULTS AND ANALYSIS

After preprocessing the data, two algorithms are applied to the supervised classification problem.

A. *Results of random forest*

We evaluate random forest algorithm and deep neural network in terms of the precision, recall, accuracy and computation time after the parameters in both algorithms are optimized.

All results were obtained when the number of trees in the forest equals 40 and the maximum depth of tree equals 15. Fig. 7 shows the prediction results for 2017 crime data classified into 3 categories. Label 1, 2, and 3 represent home, public open space and public building. Fig. 8 is the normalized results of fig. 7. The recall of label 3 is much lower compared with label 1 and 2. Table 5 shows the precision, recall and f1 scores. In my windows 10 computer with Pycharm platform, the computation time of random forest for the best accuracy performance is 94 s. The computation time is closely related to the number of trees and the depth of each tree, so it is important to consider the balance between the accuracy and the computation time.

TABLE V. CLASSIFICATION REPORT (RANDOM FOREST)

Label	Evaluation Results		
	Precision	*Recall*	*F1-score*
1	0.6531	0.6407	0.6468
2	0.5807	0.7371	0.6496
3	0.6056	0.2196	0.3223
Avg/total	0.6142	0.6109	0.5934

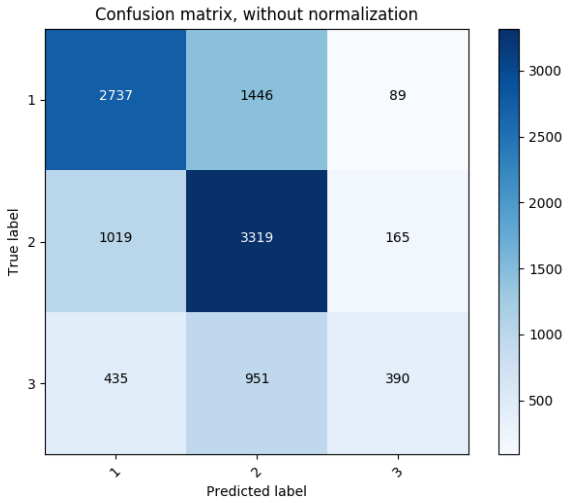

Figure 7. Prediction results for 2017 crime prediction with random forest. *(1: home; 2: public open places; 3: public buildings)*

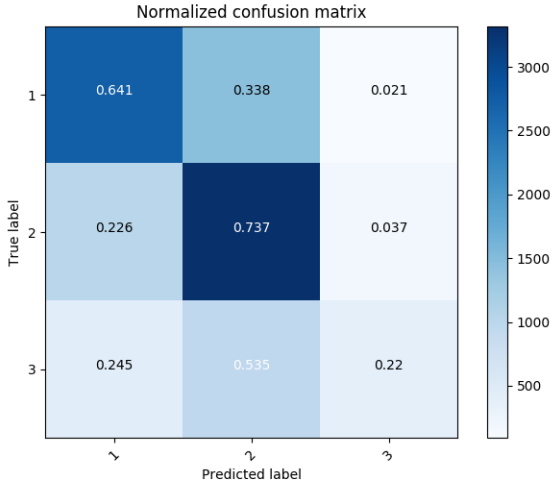

Figure 8. *Normalized prediction results* for 2017 crime prediction with random forest. *(1: home; 2: public open places; 3: public buildings)*

B. *Results of deep neural network without dense embedding*

The results of deep neural network without dense embedding are presented in this part. The one-hot coding sparse feature inputs are directly connected with the hidden layers. The dimension of input matrix is large. Table 6 indicates the performance of neural network with this structure. The computation time is 224 s with my windows 10 computer. The prediction results are presented in Fig. 9. Label 1, 2, and 3 represent home, public open space and public building. Fig. 10 is the normalized confusion matrix of fig. 9.

TABLE VI. CLASSIFICATION REPORT (DEEP NEURAL NETWORK WITHOUT DENSE EMBEDDING)

Label	Evaluation Results		
	Precision	***Recall***	***F1-score***
1	0.5684	0.6447	0.6042
2	0.5581	0.5674	0.5627
3	0.3821	0.2427	0.2968
Avg/total	0.5327	0.5440	0.5347

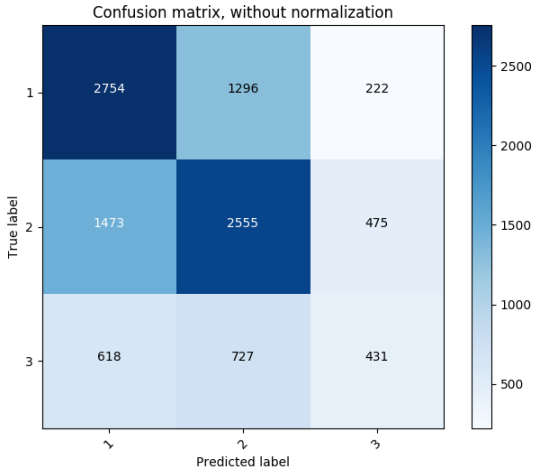

Figure 9. Prediction results for 2017 crime prediction with deep neural network without dense embeddsing. (*1: home; 2: public open places; 3: public buildings*)

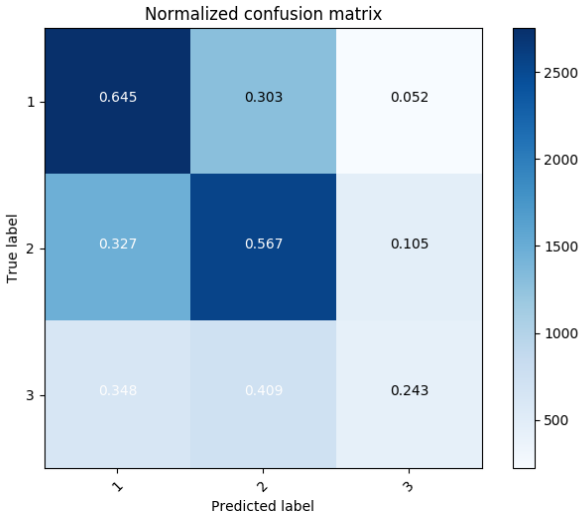

Figure 10. Normalized prediction results for 2017 crime prediction with deep neural network without dense embedding. (*1: home; 2: public open places; 3: public buildings*)

C. *Results of deep neural network with dense embedding*

Another neural network diagram is in fig. 5 with two sparse features and four non-sparse features. Four fully-connected hidden layers and 128 nodes in every layer. Activation function in all these nodes is tanh function and the dropout rate is 0.5. Fig. 11 and Fig. 12 present the

unnormalized and normalized prediction results. Table 7 displays the precision, recall and f1 scores for all the three categories. The computation time is 181 s with my windows 10 computer.

TABLE VII. CLASSIFICATION REPORT (DEEP NEURAL NETWORK WITH DENSE EMBEDDING)

Label	Evaluation Results		
	Precision	***Recall***	***F1-score***
1	0.6377	0.6664	0.6518
2	0.6052	0.6320	0.6184
3	0.4816	0.3756	0.4220
Avg/total	0.5976	0.6028	0.5988

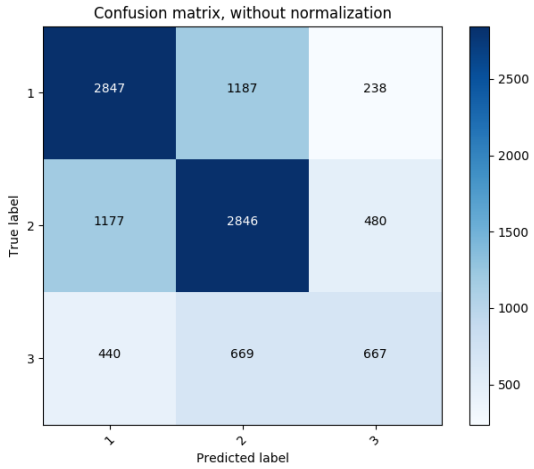

Figure 11. Prediction results for 2017 crime prediction with deep neural network with dense embeddsing. (*1: home; 2: public open places; 3: public buildings*)

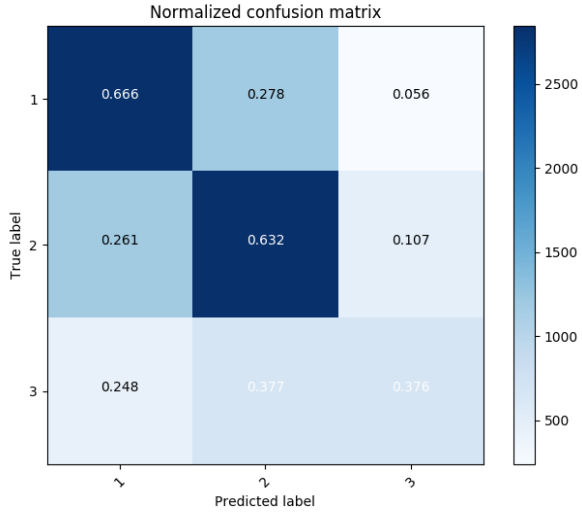

Figure 12. Normalized prediction results for 2017 crime prediction with deep neural network with dense embedding. (*1: home; 2: public open places; 3: public buildings*)

D. Comparison

For the two different neural network structures, the performance of the neural network with dense embedding is much better than the performance of the neural network without dense embedding in terms of average precision, recall and f1-score according to table 6 and table 7. The computation time of the neural network with dense embedding is shorter than the neural network without dense embedding. The comparison results indicate than dense embedding is suitable for sparse features in a neural network.

The accuracy of deep neural network with dense embedding is higher than random forest from fig. 11 and fig. 7. The precision, recall, and f1-score of random forest for label 1 and 2 is higher than deep neural network. However, the performance of deep neural network for label 3 in recall is much better than random forest, even though the performance of both algorithms in predicting label 3 is not satisfactory.

The reason for the low accuracy of prediction for label 3 might the similarity between label 2 and label 3. In the classification process for the three labels, some words in the primary location description part are ambiguous. For example, the parking lots might be open space or in a building. In order to obtain more accurate prediction results, other information, such as geometrical information, might be required to serve as the input features.

The time required to train and predict results for the deep neural network model is almost twice as that for the random forest model. In some situations, the random forest might be more suitable than neural network if the computation time is important compared with the accuracy.

VI. CONCLUSION

The problem in this paper is predicting crime location types. The input features, hour, day of a week, business day, business hour, primary crime types, and community areas, are extracted from the raw dataset. The crime location description in the raw dataset are preprocessed and classified into three prediction labels, which are home, public open space, and public buildings, before being fed into the prediction models. Random forest, deep neural network without dense embedding and with dense embedding are implemented for the prediction. Dense embedding in the deep neural network not only deceases input dimensions but also improves the performance of neural network. The models of random forest and deep neural networks are optimized. The evaluation results of optimized models indicate that deep neural network with dense embedding is the best among the three in regard to the accuracy. Otherwise, the computation time of random forest is shorter than deep neural network structure, the reason of which might be the complexity of deep neural network.

The prediction of public building label is not as accurate as other two categories with all the three models. The reason might be the unsuitable classification of the three categories

or lack of input features. The prediction accuracy might be improved by reclassification. Thus, the accuracy might be much higher if only classified into home and public place. Another strategy for improving the performance is that more datasets are applied in the prediction, such as demographic data, neighborhood appearance, and meteorological data similar to [4]. The crime occurrence is similar to our prediction. However, we focus on the prediction of location types, and it is helpful when the police patrols in the city.

REFERENCES

[1] Chandrasekar, Addarsh, Abhilash Sunder Raj, and Poorna Kumar. "Crime Prediction and Classification in San Francisco City."

[2] Vaquero Barnadas, Miquel. "Machine learning applied to crime prediction." Bachelor's thesis, Universitat Politècnica de Catalunya, 2016.

[3] Freedman, D.A. "Survival analysis: An Epidemiological hazard?". The American Statistician 62: pp.110-119, 2008. (Reprinted as Chapter 11 (pages 169–192) of Freedman (2010)).

[4] Kang, Hyeon-Woo, and Hang-Bong Kang. "Prediction of crime occurrence from multi-modal data using deep learning." PloS one 12.4, 2017: e0176244.

[5] Wang, Tong and Rudin, Cynthia and Wagner, Daniel and Sevieri, Rich. "Machine Learning and Knowledge Discovery in Databases" pp.515-530, 2013.

[6] Adepeju, Monsuru, Gabriel Rosser, and Tao Cheng. "Novel evaluation metrics for sparse spatio-temporal point process hotspot predictions-a crime case study." *International Journal of Geographical Information Science* 30.11, pp.2133-2154, 2016.

[7] Mohler, George. "Marked point process hotspot maps for homicide and gun crime prediction in Chicago." *International Journal of Forecasting* 30.3, pp.491-497, 2014.

[8] Iqbal, Rizwan, et al. "An experimental study of classification algorithms for crime prediction." *Indian Journal of Science and Technology* 6.3, pp.4219-4225, 2013.

[9] McClendon, Lawrence, and Natarajan Meghanathan. "Using Machine Learning Algorithms to Analyze Crime Data." *Machine Learning and Applications: An International Journal (MLAIJ)* 2.1 2015.

[10] R. William Adderley, "The use of data mining techniques in crime trend analysis and offender profiling," Ph.D. thesis, University of Wolverhampton, Wolverhampton, England , 2007.

[11] Keyvanpour, Mohammad Reza, Mostafa Javideh, and Mohammad Reza Ebrahimi. "Detecting and investigating crime by means of data mining: a general crime matching framework." *Procedia Computer Science* 3, pp.872-880, 2011.

[12] Gerber, Matthew S. "Predicting crime using Twitter and kernel density estimation." *Decision Support Systems* 61, pp.115-125,2014.

[13] Kianmehr, Keivan, and Reda Alhajj. "Effectiveness of support vector machine for crime hot-spots prediction." *Applied Artificial Intelligence* 22.5, pp.433-458, 2008.

[14] Mohler, George. "Marked point process hotspot maps for homicide and gun crime prediction in Chicago." *International Journal of Forecasting* 30.3, pp.491-497, 2014.

[15] Noor, Noor Maizura Mohamad, et al. "A review on a classification framework for supporting decision making in crime prevention." *Journal of Artificial Intelligence* 8.1, pp.17, 2015.

[16] Zhang, Weinan, Tianming Du, and Jun Wang. "Deep learning over multi-field categorical data." *European conference on information retrieval*. Springer International Publishing, 2016.

[17] Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.* ACM, 2016.

[18] Guo, Huifeng, et al. "DeepFM: A Factorization-Machine based Neural Network for CTR Prediction." arXiv preprint arXiv:1703.04247, 2017.

[19] Breiman, Leo. "Random forests." *Machine learning* 45.1, 5-32, 2001.

[20] Rodriguez-Galiano, Victor Francisco, et al. "An assessment of the effectiveness of a random forest classifier for land-cover classification." *ISPRS Journal of Photogrammetry and Remote Sensing* 67, pp.93-104, 2012.

[21] Wang, Aiping, et al. "An incremental extremely random forest classifier for online learning and tracking." *Image Processing (ICIP), 2009 16th IEEE International Conference on. IEEE*, 2009.

[22] Svetnik, Vladimir, et al. "Random forest: a classification and regression tool for compound classification and QSAR modeling." *Journal of chemical information and computer sciences* 43.6, pp.1947-195, 8, 2003.

[23] Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep struc-tured semantic models for web search using clickthrough data. In: CIKM. pp. 2333–2338, 2013.

[24] Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: CIKM 2014.

[25] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." *Acoustics, speech and signal processing (icassp)*, 2013 ieee international conference on. IEEE, 2013.

[26] Olson, David L.; and Delen, Dursun; Advanced Data Mining Techniques, Springer, 1st edition, page 138, ISBN 3-540-76916-1, February 1, 2008.

[27] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[28] Gal, Yarin, and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks." *Advances in neural information processing systems*. 2016.

[29] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research* 15.1, 1929-1958, 2014.

[30] Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv: 1207.0580, 2012.

[31] Larochelle, Hugo, et al. "Exploring strategies for training deep neural networks." *Journal of Machine Learning Research*, 1-40, 10.Jan, 2009.