Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 9922162, 12 pages
https://doi.org/10.1155/2023/9922162

Research Article

Fast Prediction for Criminal Suspects through Neighbor Mutual Information-Based Latent Network

Jong Ho Jhee,[1] **Myung Jun Kim**,[2] **Myeonggeon Park**,[3] **Jeongheun Yeon**,[4] **and Hyunjung Shin**[4,5]

[1]*Center for KIURI Bio-Artificial Intelligence, Ajou University School of Medicine, Suwon 16499, Republic of Korea*
[2]*Soda, INRIA Saclay, Palaiseau 91120, France*
[3]*Marketboro Corp., Seongnam 13488, Republic of Korea*
[4]*Department of Artificial Intelligence, Ajou University, Suwon 16499, Republic of Korea*
[5]*Department of Industrial Engineering, Ajou University, Suwon 16499, Republic of Korea*

Correspondence should be addressed to Hyunjung Shin; shin@ajou.ac.kr

Received 12 April 2023; Revised 28 June 2023; Accepted 20 September 2023; Published 6 October 2023

Academic Editor: Vittorio Memmolo

One of the interesting characteristics of crime data is that criminal cases are often interrelated. Criminal acts may be similar, and similar incidents may occur consecutively by the same offender or by the same criminal group. Among many machine learning algorithms, network-based approaches are well-suited to reflect these associative characteristics. Applying machine learning to criminal networks composed of cases and their associates can predict potential suspects. This narrows the scope of an investigation, saving time and cost. However, inference from criminal networks is not straightforward as it requires being able to process complex information entangled with case-to-case, person-to-person, and case-to-person connections. Besides, being useful at a crime scene requires urgency. However, predictions from network-based machine learning algorithms are generally slow when the data is large and complex in structure. These limitations are an immediate barrier to any practical use of the criminal network geared by machine learning. In this study, we propose a criminal network-based suspect prediction framework. The network we designed has a unique structure, such as a sandwich panel, in which one side is a network of crime cases and the other side is a network of people such as victims, criminals, and witnesses. The two networks are connected by relationships between the case and the persons involved in the case. The proposed method is then further developed into a fast inference algorithm for large-scale criminal networks. Experiments on benchmark data showed that the fast inference algorithm significantly reduced execution time while still being competitive in performance comparisons of the original algorithm and other existing approaches. Based on actual crime data provided by the Korean National Police, several examples of how the proposed method is applied are shown.

1. Introduction

As the number of criminal cases continues to rise, there has been a notable increase in the utilization of machine learning (ML) for criminal investigation support. Various areas, including crime pattern analysis [1–3], fraud detection, traffic violation monitoring, sexual assault investigations, and cybercrime analysis [4–6], have seen the application of ML techniques. While challenges related to data confidentiality still exist, the potential of machine learning in aiding case investigations is widely recognized. The valuable insights derived from accumulated criminal cases play a crucial role in providing clues and assisting law enforcement agencies in their investigative efforts.

In the meantime, one of the characteristics of crime data is that criminal cases are related. Criminal behaviors may be similar, and similar incidents may occur consecutively by the same offender or by the same criminal group. To study crime cases, social network approaches have emerged that reflect the associations between people, and these network-based (or graph-based) methods are expected to become one of the standard tools in criminology research [7–9]. There are early works applied to criminal investigations using network-based ML algorithms. Weber et al. [10] applied a graph convolutional network algorithm to financial data for antimoney laundering forensic analysis. The task was to predict the suspiciousness of a given target, and Das et al. [11] used graph-based clustering to extract relational information from crime data of India. The named entities were extracted from the text corpus of crimes and converted into vectors using the word2vec algorithm [9]. The network was then constructed by measuring the similarity between these vectors. Clusters found in the network represent incidents or offenders with similar patterns. Meanwhile, social networks of online auction users are used for fraud detection. By labeling known scammers and legitimate users, a label propagation algorithm is used to predict potential scammers [12]. Also, in [13], an online advertising network was constructed and analyzed using Laplacian SVM to detect human trafficking advertisements. Khan et al. [14] proposed a crime prediction model by comparing three known algorithms, Naïve Bayes, random forest, and gradient boosting decision tree, and classified the top ten crimes from the San Francisco crime data. A combined framework of graph representation learning and machine learning methods is introduced to predict the amount of money exchanged among criminal agents and to recover the missing criminal partnerships [15]. Meanwhile, one of the important parts in criminal network analysis is the link prediction problem. In many cases, there is a possibility of acquiring missing or incomplete information by the crime investigation, and one might want to recover missing links or connections among individuals or resources in the information. In that sense, Berlusconi et al. [16] proposed a method to identify missing links in a criminal network by classifying links based on the topological analysis and applied it to the Italian criminal case data against a mafia group. Also, to make the link prediction robust to varying relations in a criminal network, Calderoni et al. [17] applied various link prediction algorithms and observed the algorithm that leverages the full graph topology.

Of the many network-based ML algorithms, the graph-based semisupervised learning (GSSL) algorithm is one of the most popular because it is easy to use, can handle situations where data have few labels, and its inference is intuitive along the network structure [18–24]. Therefore, the scope of application is wide where relational information is important, such as finding key genes using disease and gene networks [25], predicting protein functions using multiple biological networks [26], and classifying historical figures to political parties using many relationships such as blood ties, academic ties, and geographic proximity [27, 28]. In the domain of social networks, GSSL is used to create relevant links to the concept referred in Wikipedia for all tweet mentions [29], and in other applications, it is used to detect fake users from a large volume of Twitter networks [30]. In the computer vision domain, GSSL is employed for hyperspectral image classification based on image networks [31], and in the natural language processing domain, part-of-speech tagging was performed by applying GSSL on a random field network [32].

In this study, we propose a framework for predicting suspect candidates by applying GSSL to criminal networks. The network is designed to be layered, like a sandwich panel, with a network of crime cases on one side and a network of people, such as victims, offenders, and witnesses, on the other side. Nodes or entities in each network are connected through similarities. It is also connected to nodes belonging to the other network reflecting the relationships between the case and the person who is involved in the case. Meanwhile, applying GSSL to the crime scene requires agility to achieve immediate results. However, when the data size is large, the time complexity of network-based algorithms increases exponentially according to the size of the network, so the inference speed by GSSL is inevitably slowed down. Given that crime scenes always demand a sense of urgency, the slow inference is fatal indeed to solving cases. Therefore, we propose a fast GSSL algorithm for large-scale criminal networks to mitigate the limitation. The idea is to insert a latent network of cluster centroids and link cases or persons to the corresponding centroids. Exhaustive searches are avoided by inserting a network between the network of cases and the network of persons. The scope of the search is reduced to only a small set of nodes (cases or people) belonging to the same cluster in the latent network. On the other hand, high-dimensional data are newly represented as low-dimensional vectors by using the clusters in the latent network. The proposed method is called neighbor mutual information semisupervised learning (MISSL) because it uses the mutual information between the clusters and the neighbors of the nodes. MISSL is robust to nonspherical clusters of various sizes and shapes. The number of clusters does not increase linearly with the number of cases or people, which is advantageous in terms of memory efficiency, especially for networks that require frequent updates and are constantly growing.

The remaining sections of the paper are organized as follows. Section 2 describes the method of building the criminal network and explains the prediction procedure using GSSL. Section 3 describes how to insert the latent network and details the fast prediction algorithm, MISSL. Section 4 demonstrates the comparative experiments on benchmarking datasets, and Section 5 presents a practical application of criminal network analysis. Finally, conclusions are drawn in Section 6.

2. Criminal Network and Suspect Scoring

2.1. Network Construction. Criminal acts often exhibit similarities, and it is not uncommon for similar incidents to occur consecutively, either by the same offender or by a particular criminal group. In addition, in some cases, the patterns of crimes may bear resemblance, even if the offenders involved are different, resembling a copycat scenario seen in serial murder cases. Considering these factors, a network serves as an effective tool for illustrating connections between crimes and people. The criminal network we have developed captures associations between cases, individuals, and interactions between cases and individuals. This network follows a two-layered structure, with one layer representing the network of crime cases and the other layer representing the network of people. These two layers are interconnected through relationships that link specific cases with the corresponding individuals involved.

Let the criminal network be denoted as $G = (G^p \times G^c)$ where G^p stands for the people network and G^c stands for the case network. Each network is represented as $G = (V, W)$ where $V = \{x_i\}_{i=1}^{n}$, $x_i \in \mathbb{R}^d$ is a set of nodes, and $W = \{w_{ij}\}_{i,j=1}^{n}$ is a set of weighted edges. The weight w_{ij} is determined by the similarity between nodes x_i and x_j. Generally, the Gaussian kernels are widely used for similarity calculation and are represented as follows:

$$w_{ij} = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right), \tag{1}$$

where σ is a bandwidth parameter. To calculate the similarity w_{ij}^{p} in the people network G^p, the demographic information, such as age, gender, address, occupation, and criminal information, such as the history of criminal records, are considered as input features. For the w_{ij}^{c} of the case network G^c, crime reports including the location of an incident are used to measure the similarity between crime cases. In particular, to calculate the similarity between text-type crime reports, each report was converted into a term vector through term frequency-inverse document frequency (TF-IDF) [33] (more details are explained in Section 3). The similarity between a person and a case is denoted as w_{ij}^{pc} where $x_i \in V^p$ and $x_j \in V^c$. If a person is involved in a certain criminal case, then the person is connected to the case. Unlike other similarity weights in the network, the edge weight w_{ij}^{pc} is set to "1" (connected) if the person is involved in the case as a suspect, a victim, or a witness and "0" (disconnected) if the person is not involved in the case. The left side of Figure 1 shows the schematic picture of the criminal network.

2.2. Crime Data. The crime data from January 2000 to December 2019 are collected from the Korea Information System of Criminal Justice Services (KICS) with the help of the Police Science Institute, Asan, Korea [34]. The types of crimes include a battery, assault, drug, theft, traffic violation, disorderly conduct, and financial crime. The data can be categorized into people data and crime case data. The data of people contain personal information about the suspects, victims, and witnesses. In addition, the information of past criminal history and possession of firearms appears if they exist. The crime case data contains the date, location, type of crime, and the case summary report written by the officer in charge. Usually, a suspect, a victim, and a witness are related to a single case. Some cases have multiple victims or witnesses and also there are cases where a suspect from one case appears in another.

Among the crime case data variables, the summary report is unstructured data in text format. Since most of the information about the incident is included in this report, preprocessing of the text is vital. In order to process text data, nouns and verbs were extracted using the Korean morphology analyzer KoNLPy [35]. It is known as one of the best analyzers among open-source Korean morphology analyzers. Extracted words from the reports are converted into vectors through TF-IDF [33]. By using TF-IDF, the influence of unnecessarily repeated words can be reduced to some extent, and important information can be highlighted. Figure 1 shows an example of a part of the case report, and more details are described in Table 1.

We built a criminal network with a network of 43,603 people and 20,500 cases. Both networks are sparsely connected, that is, the network densities are 2.49% and 2.90%, respectively. To overcome spare connections, the aforementioned link prediction algorithms can be used to infer missing links among suspects, victims, or witnesses to reinforce the criminal network [16, 17].

2.3. Suspect Scoring from the Criminal Network. The primary objective of a criminal network is to predict potential suspect candidates when a new crime case emerges, that is, identifying individuals who are likely to be involved. Suspect scores are calculated for each node in the network of people, and those with the highest scores are recommended as potential candidates. However, in the immediate aftermath of a criminal case, it is often the case where only a few individuals, such as the victim or a small number of people are directly involved, have the knowledge of the incident. Consequently, the available labeled data for training the predictive model is extremely limited, resulting in a sparse dataset. Therefore, GSSL [18] was adopted since it can learn even with just one label and further developed to fit the hierarchical structure of the criminal network.

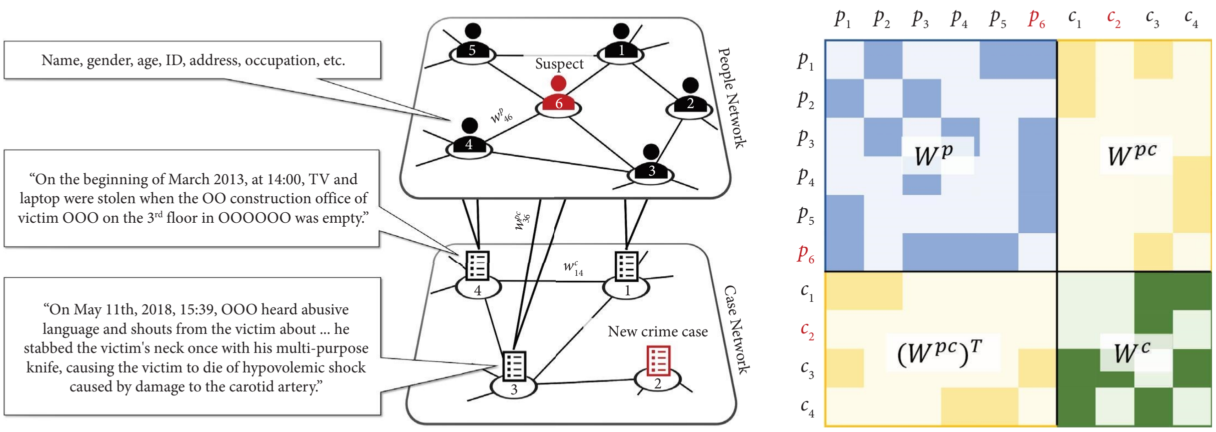
Given a weight matrix $W = \{w_{ij}\}_{i,j=1}^{n}$ of a plain (not layered) network G, the Laplacian is defined as $L = D - W$ where diagonal matrix $D = \operatorname{diag}(d_i)$ and $d_i = \sum_{j=1}^{n} w_{ij}$. Then, simple GSSL works by optimizing the objective function are defined as

$$\min_{\mathbf{f}} \ (f - y)^T (f - y) + \mu f^T \mathrm{Lf}. \tag{2}$$

Equation (2) minimizes the loss between predicted labels $f = (f_1, \ldots, f_l, f_{l+1}, \ldots, f_{l+u})^T$ and node labels $y = (y_1, \ldots, y_l, 0, \ldots, 0)^T$, while smoothing the neighbor nodes' labels to be similar, i.e., GSSL propagates the node labels to unlabeled nodes through weighted edges in the network. The solution of (2) can be obtained as

$$f = (I + \mu L)^{-1} y, \tag{3}$$

where the parameter $\mu \geq 0$ controls the tradeoff between loss and smoothness. This learning framework for a plain network is straightforward to extend to a layered network. Suppose the layered network has a total of $N = n_p + n_c$ nodes



Name, gender, age, ID, address, occupation, etc.

"On the beginning of March 2013, at 14:00, TV and laptop were stolen when the OO construction office of victim OOO on the 3rd floor in OOOOOO was empty."

"On May 11th, 2018, 15:39, OOO heard abusive language and shouts from the victim about ... he stabbed the victim's neck once with his multi-purpose knife, causing the victim to die of hypovolemic shock caused by damage to the carotid artery."

Figure 1: The schematic picture of the criminal network. The upper layer is the people network and the lower layer is the case network. The corresponding weight matrices are represented as W^p and W^c in the figure on the right, respectively. Two networks are connected by person-to-case edges denoted by W^{pc} in the weight matrix. A person node has demographic information of an individual while a case node has information on crime reports (texts are translated from Korean to English. Names or titles in texts are de-identified. All the meaningless symbols were removed).

Table 1: Crime data description.

Variables		Descriptions
Person	Basic info	Nationality, gender (female or male), age, height, weight, address, and occupation
	History	History of criminal records, possession of a firearm, and sexual assault or not
	Class	Suspect/victim/witness
Crime case	Basic info	Time of incidence, location, type of crime, and arrested or not
	Case summary	Crime summary report (text)
	Type	Battery, assault, drug, theft, traffic violation, disorderly conduct, and financial crime

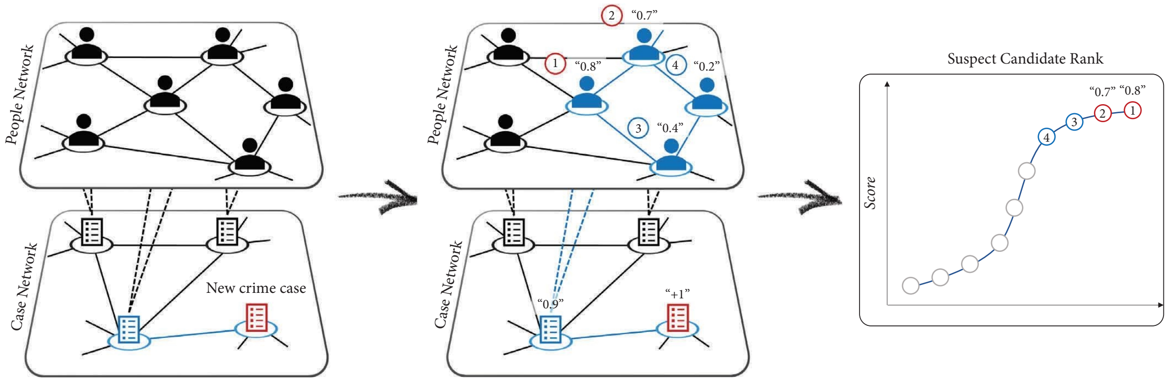
and a weight matrix of W consisting of W^p, W^c, and W^{pc} as shown on the right side of Figure 1, then by simply substituting the Laplacian of (2) for the plain network to that of the layered network, the solution is obtained by (3) for the layered network. If there are many layers, i.e., more than two layers are structured hierarchically, then calculating (3) is computationally highly demanding because the weight matrix is huge. In this case, approximation methods such as the Woodbury formula or the Nystrom method can be used to speed up the algorithm [36]. The algorithm is applied to predict the suspect candidates from a small set of criminal networks [37].

Now if case i is known, we can calculate the suspect scores. By setting the label of the i^{th} case node as $y = (0, \ldots, 0, y_i = 1, 0, \ldots, 0)^T$, the label influence propagates to both the case network and the people network, eventually computing (3). People with high f scores are regarded as the suspect candidates. The process of suspect scoring is shown in Figure 2. First, when a new case marked in red in Figure 2(a) comes in, it is connected to the most similar existing cases. Assigning the label "1" to the new case (zeros to the remaining nodes) propagates the label to similar cases in the vicinity, and it spreads to people involved in these similar cases through the blue edge that bridges the case with people in the upper layer (Figure 2(b)). Finally, people are ranked in descending order of score (Figure 2(c)). The questionable suspects appear as those with the highest scores (circled in red).

3. Fast Criminal Network-Based Suspect Prediction

Scoring suspect candidates using GSSL works well if the network is reasonably sized. However, it slows down as the number of crimes and the number of people increases, making it more likely to be lethal at a crime scene where urgent predictions are needed to solve the case. The slowdown is mainly attributed to memory and time-consuming computations when retrieving similar cases. When a new case comes in, an exhaustive search of existing cases is conducted to find the most similar case. To make matters worse, every case is a high-dimensional text vector extracted from criminal case reports. For reference, the number of cases is over 20,000, the number of related people is over 100,000, and the dimensionality of a text vector is over 2,000. To alleviate the difficulty, we propose a new search that reduces the search scope from global to local, and a new representation of text vectors that drastically reduces the dimensionality from hundreds or thousands to a few dimensions.

3.1. Local Search via the Latent Network. Instead of an exhaustive search, the idea is to insert a latent network between the case network and the people network. The latent network is composed of centroids of clusters that are organized in advance. When a new case comes in, it calculates its



(a) (b) (c)

Figure 2: The process of suspect candidate scoring on the criminal network. (a) The new crime case (red) and its similar cases in the vicinity. (b) Label propagation from the case nodes to people nodes in the upper layer. (c) Scoring people in descending order and suspect candidates with the highest scores.

similarities to the centroids and selects the closest cluster. Then, the search is narrowed down to only a small set of case nodes that belong to the same cluster to which the new case belongs by referencing the latent network. The process is depicted in Figure 3(a). Therefore, the role of the latent network is important. In other words, clustering results matter. However, real data are not uniformly distributed or spherically shaped such as a normal distribution, as most well-known clustering techniques assume. Figure 3(b) illustrates clusters with different shapes and varying densities. As a way to overcome this limitation, we suggest measuring the relative location of a data point by looking around its neighbors and their cluster memberships. That is, if the cluster memberships of its neighbors are homogeneous (belong to the same cluster), then the data point is likely to be at the core of the cluster. Conversely, if the cluster memberships of its neighbors are heterogeneous (some belong to one cluster and others belong to another cluster), then the data point is likely to be located on the boundaries of clusters. Figure 3(b) exemplifies the estimation of the relative location of data points. For x_1, the neighbors are homogeneous in terms of cluster membership, so it is estimated to be located close to the core of cluster 1, whereas x_2 or x_3 is heterogeneous in cluster membership of its neighbors, so they are estimated to be near the cluster boundaries. This concept allows the cluster regions to be well-defined even when the data distribution is not spherical and the densities change.

3.2. Latent Dimension via Neighbor Mutual Information. The clusters are reused to reduce a higher dimension to a lower dimension. Hereafter, the resulting dimension is denoted as a latent dimension. The size of the latent dimension is determined by the number of clusters, and the value of each dimension is determined by the degree to which the data point belongs to each cluster. Therefore, d-dimensional vectors are reduced to m-dimensional vectors that are much smaller than the original dimension ($d \gg m$), and the latent values are measured by neighbor mutual information (NMI), which has been proposed here.

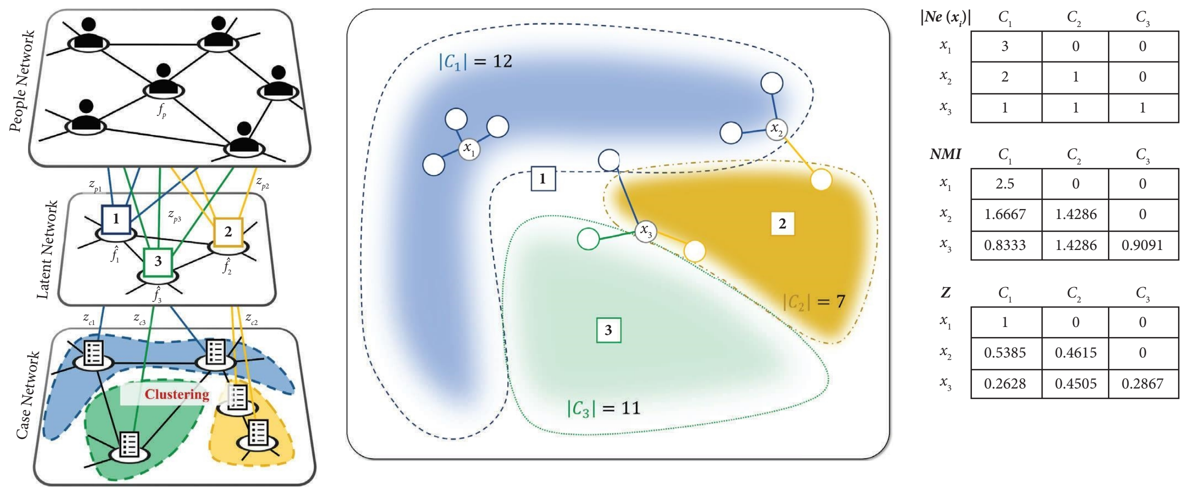
The NMI of a data point (node) is the amount of mutual information between the clusters and the neighbors of the data point. Generally, for a pair of discrete random variables X and Y, mutual information is defined as $\sum_{x\in X}\sum_{y\in Y} p(x, y)\log p(x, y)/p(x)p(y)$, where $p(x, y)$ is the joint probability [38, 39]. If $p(x, y) = p(x)p(y)$, then the logarithm becomes zero and thus mutual information becomes zero, which means X and Y are independent. Analogously, the NMI of node x_i is defined between the cluster $C_m \in \{C_1, \ldots, C_M\}$ and the set of its neighbors $\mathrm{Ne}(x_i)$ as

$$\mathrm{NMI}(x_i, C_m) = \frac{|\mathrm{Ne}(x_i) \cap C_m|/N}{|\mathrm{Ne}(x_i)|/N \cdot |C_m|/N} = \frac{N \cdot |\mathrm{Ne}(x_i) \cap C_m|}{|\mathrm{Ne}(x_i)| \cdot |C_m|}, \tag{4}$$

where $|\cdot|$ denotes cardinality and N denotes the total number of nodes in the network. $|C_m|$ is the number of nodes in cluster C_m and $|\mathrm{Ne}(x_i)|$ is the number of neighbor nodes of x_i. Thus, $|\mathrm{Ne}(x_i) \cap C_m|$ is the number of the neighbor nodes of x_i belonging to cluster C_m. The NMI increases when more neighbor nodes belong to the cluster and vice versa. In the extreme case, when all the neighbor nodes are members of one single cluster ($|C_m| = N$), then $\mathrm{NMI}(x_i, C_m) = N \cdot |\mathrm{Ne}(x_i)|/|\mathrm{Ne}(x_i)| \cdot |C_m| = 1$, whereas when none of the neighbor nodes belong to the cluster, then, $\mathrm{NMI}(x_i, C_m) = 0$. Finally, z_{ij} represents the normalized version of $\mathrm{NMI}(x_i, C_m)$, which satisfies the nonnegativity and sum to one constraint.

$$z_{im} = \frac{\mathrm{NMI}(x_i, C_m)}{\sum_m \mathrm{NMI}(x_i, C_m)}. \tag{5}$$

Figures 3(b) and 3(c) show several examples of NMIs that transform x_i to the relative locations between clusters. For node x_1, $\mathrm{NMI}(x_1, C_1) = (30 \times 3)/(3 \times 12) = 2.5$ by (4) since C_1 has 12 nodes and all three neighbors are a part of the cluster, whereas $\mathrm{NMI}(x_1, C_2)$ and $\mathrm{NMI}(x_1, C_3)$ are zeros. Thus, $z_1 = [1, 0, 0]$. Note that, node x_1 is in the core of C_1 far from C_2 and C_3. Conversely, node x_3 is located on the boundary of C_2 and has neighbors belonging to C_1, C_2, and



$\|Ne(x_i)\|$	C_1	C_2	C_3
x_1	3	0	0
x_2	2	1	0
x_3	1	1	1

NMI	C_1	C_2	C_3
x_1	2.5	0	0
x_2	1.6667	1.4286	0
x_3	0.8333	1.4286	0.9091

Z	C_1	C_2	C_3
x_1	1	0	0
x_2	0.5385	0.4615	0
x_3	0.2628	0.4505	0.2867

(a) (b) (c)

Figure 3: The latent network and the latent vectors. (a) A criminal network with the latent network composed of centroids of the case clusters. Colors represent cluster membership and squares represent cluster centroids. (b) The relative location estimation in circumstances of clusters of different shapes and densities. x_1 is estimated to be near the core of cluster 1 because all three of its neighbors belong to cluster 1 (homogeneous in terms of cluster membership), whereas x_2 or x_3 is estimated to be near the boundaries of clusters because the neighbors differ in cluster membership (heterogeneous). (c) The cluster memberships of x_1, $x_{2,}$ and x_3, NMI calculation result, and their latent vector Z.

C_3, respectively. Its normalized NMI is $z_3 = [0.26{,}0.45{,}0.29]$. Note that, z_3 has nonzero values for all three clusters but weighs C_2 the most since it belongs to the cluster. The right bottom of Figure 3(c) represents matrix $Z \in \mathbb{R}^{N\times M}$ of latent vectors obtained by (5). The dimension values of z_i give the relative location of node x_i using M clusters. In Figure 3(a), z_i is depicted as M edges connecting x_i to M clusters.

There are tradeoffs between the number of clusters and the latent dimension. If the number of clusters increases, the number of centroids in the latent network increases, and it leads to a reduction in the scope of local search because the data are split into more clusters and thus fewer data in each cluster. However, the latent dimension increases, that is, increasing the number of clusters affects both increasing and decreasing computation time and vice versa. Therefore, it is necessary to adjust the number of clusters according to the situation at hand, so as to figure out whether search coverage or dimensionality reduction is more important.

3.3. *Scoring via the Latent Network.*

The latent network makes the computation of node scoring for the case network much lighter. Given a weight matrix $\widehat{W} = \{\widehat{w}_{ij}\}_{i,j=1}^{M}$ between M centroids, a score vector $\widehat{f} = (\widehat{f}_1, \ldots, \widehat{f}_M)^T$ can be calculated similarly to (3). It is now straightforward to derive a score vector $f_i = z_i^T \widehat{f}$, which is the sum of the centroid scores $\widehat{f}$ weighted by the latent vector z_i. Therefore, the scoring problems (2) and (3) turn into finding the optimal centroid scores on the latent network. The objective function is represented as follows:

$$\min_{\widehat{f}} \, (Z\widehat{f} - y)^T (Z\widehat{f} - y) + \mu \widehat{f}^T \widehat{L} \widehat{f}, \tag{6}$$

where $\widehat{L} = Z^T L Z \in \mathbb{R}^{M\times M}$, i.e., the graph Laplacian of the latent network. The only difference from (2) is that the loss is computed using the weighted sum of centroid scores and the smoothness is optimized on the latent network. It is thus trivial to derive the solution to (6).

$$\widehat{f} = \left(Z^T Z + \mu \widehat{L}\right)^{-1} Z^T y. \tag{7}$$

Finally, the score vector f is predicted using the centroid scores $\widehat{f}$ and the latent representation, Z, that is,

$$f = \frac{Z\widehat{f}}{\mathbf{1}^T Z \widehat{f}}, \tag{8}$$

where $\mathbf{1}^T Z\widehat{f}$ is the normalization.

From now on, the proposed method is named as MISSL, that is, the abbreviation of neighborhood information-based semisupervised learning. Constructing the latent network takes an extra $O(\mathrm{NM}^2)$, but since M is small, it is not a huge burden. However, once the latent network is constructed, it provides significant advantages for computing the inverse matrix of the solution. For the latent network in (7), it has $O(M^3)$ complexity, whereas the original network in (3) has $O(N^3)$ complexity ($N \gg M$). Also, MISSL was originally designed to work on undirected networks. However, it can be extended to directed networks by converting asymmetric matrices to the symmetric graph Laplacian [40].

3.4. *Application to Criminal Network.*

Applying MISSL to criminal networks is simple. Conceptually, a latent network of centroids lies between the people network and the case network, connecting the two networks. Edges from a centroid node to case nodes are connected via the latent vector, as explained in the previous section. When a new case comes in, MISSL finds the centroid of the nearest cluster C_m, and converts node x_{new} into a new latent vector using only the representation matrix for that cluster. More precisely, we first calculate the similarity between centroids and the node

Table 2: Benchmark datasets.

Dataset	Classes	Dimension	Nodes	Remark
g241c	2	241	1,500	Artificial
g241n	2	241	1,500	Artificial
MNIST	10	784	70,000	Large-scale
CIFAR-10	10	3,072	60,000	Large-scale

x_{new} to find the nearest cluster. Within the cluster C_m, mixing weight ω is optimized to transform x_{new} to z_{new} and the latent vector Z is updated. Then, we applied MISSL for the new case using the updated latent vector. Conversion from x_{new} to z_{new} is performed by finding a mixing weight $\omega \in \mathbb{R}^{M\times d}$ which is obtained by solving the following problem:

$$\min_{\omega} \ (Z_m\omega - X_m)^T (Z_m\omega - X_m), \tag{9}$$

where $X_m \in \mathbb{R}^{N_m\times d}$ and $Z_m \in \mathbb{R}^{N_m\times M}$ are the matrices of the data points belonging to cluster C_m. From this, the optimal weight for conversion is calculated as $\omega = (Z_m^T Z_m)^{-1} Z_m^T X_m$. So, the latent vector for the new case is

$$z_{\text{new}} = \omega x_{\text{new}} = \left[\left(Z_m^T Z_m\right)^{-1} Z_m^T X_m\right] x_{\text{new}}. \tag{10}$$

The new latent vector is connected to the case network by finding the closest cases. MISSL then scores suspect candidates in the people network according to (7). By using MISSL, a much faster prediction is possible than the naïve SSL in Section 2. This allows official crime investigators to quickly weed out suspect candidates at a crime scene.

4. Experiments on Benchmark Data

Experimental results in the following section show that the proposed algorithm has a fast inference time and competitive performance compared to the existing approaches.

4.1. Data. We evaluated the proposed method, MISSL, on benchmark datasets. The datasets are g241c, g241n, MNIST, and CIFAR-10. g241c and g241n datasets were artificially generated to hold the cluster assumption. The data points of g241c were drawn from each of the two unit-variance isotropic Gaussians. The label of a data point represents the Gaussian it was drawn from. The data points of g241n were drawn from each of the two unit-variance isotropic Gaussians which have a potentially misleading cluster structure and no manifold structure. The centers for the positive class have a distance of six in a random direction and the centers for the negative class were fixed by moving from the former centers to a distance of 2.5 in a random direction. All dimensions were standardized to zero mean and unit variance. The number of dimensions and data points are 241 and 1,500, respectively [41]. MNIST is a handwritten digit dataset of 28 by 28 grayscale normalized and centered images. The labels of the dataset contain the number from zero to nine and the number of data points is 70,000 [42]. The CIFAR-10 dataset (Canadian Institute for Advanced Research) is a subset of the 80 million tiny images. The CIFAR-10 dataset contains 60,000 32 × 32 color images in 10 different classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class [43]. Table 2 summarizes the datasets.

4.2. Experimental Setup. We compared the performance and scalability of MISSL with another scalable method called anchor graph regularization (AGR) [44]. Here, an anchor of AGR that employs RBF kernel or local linear embedding corresponds to a centroid of our method [45]. The two network-based methods use anchors or centroids to reduce the heavy computation of GSSL. The performance of Naïve GSSL was used as a reference. This indicates that both methods, MISSL and AGR, cannot overwhelm the baseline performance. The dataset was divided into the 20-cross validation (20-cv) format. However, unlike the supervised learning setting, SSL assumes a few labeled data, so a single set is the training set, and the remaining 19 sets are configured as the validation sets; that is, the labeled data is 5% (positive: 2.5% and negative: 2.5%) of the entire dataset. The experiment was repeated 10 times to optimize the hyperparameters.

4.3. Results. Table 3 shows the performance results on benchmark datasets measured with the area under the receiver operating characteristic curve (AUC), along with the hyperparameters of AGR, m (the number of centroids) and r (the number of nearest centroids), and hyperparameters of MISSL, m (the number of clusters) and k (the number of nearest neighbors). The average performances over 20-cv of AGR and MISSL are 97.65% and 98.98% of the reference performance, respectively. MISSL showed better performance than AGR in all datasets. It shows that the latent dimension using mutual information of MISSL better represents data than RBF kernels or LLE of AGR.

The experimental results on computation time are shown in Table 4. The computation time was measured separately in three parts: the network construction (or clustering), the latent representation, and the inference (prediction). For GSSL, no conversion time for new representation is required, and clustering time is only required for AGR and MISSL. However, GSSL takes a lot of time for both network construction and inference,

Table 3: AUC comparison on benchmark datasets.

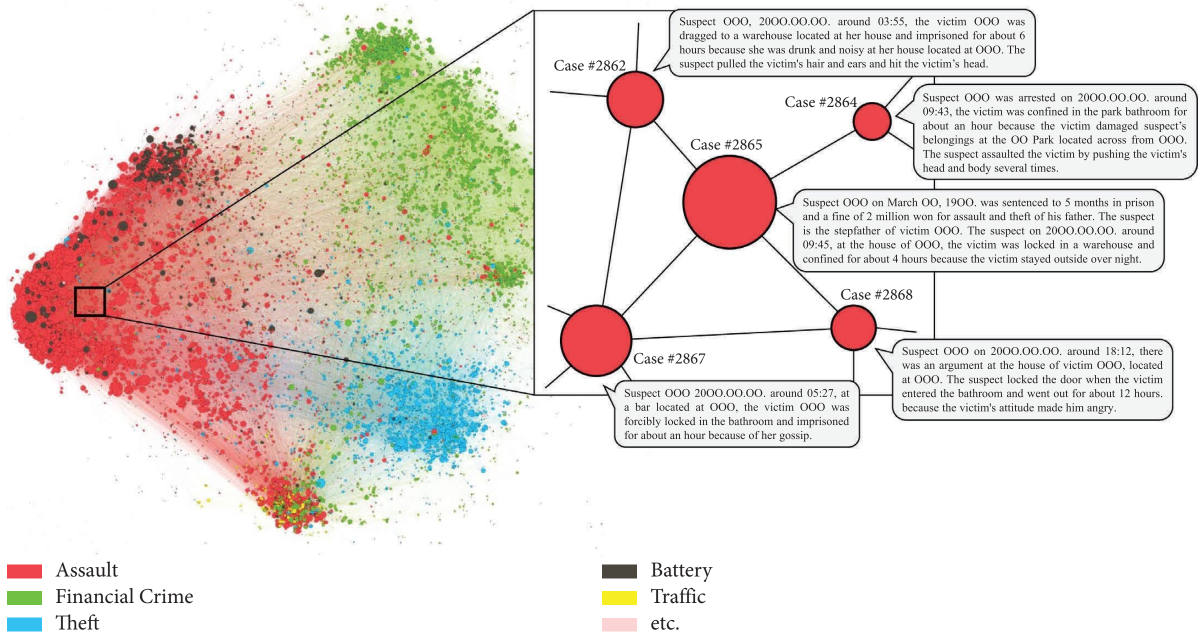
Method	g241c	g241n	MNIST	CIFAR-10
AGR	0.6438 ± 0.0288	0.6448 ± 0.0468	0.9876 ± 0.0002	0.7305 ± 0.0014
(m, r)	(150, 5)	(150, 5)	(300, 10)	(300, 10)
MISSL	**0.7652 ± 0.0266**	**0.6744 ± 0.0257**	**0.9885 ± 0.0006**	**0.7324 ± 0.0022**
(m, k)	(150, 10)	(150, 10)	(300, 20)	(300, 20)
GSSL	0.7808 ± 0.01890	0.7504 ± 0.0682	0.9896 ± 0.0003	0.7340 ± 0.0025

The bold values refer to the values of the proposed method (MISSL).

Table 4: Computation time (sec.) on benchmark datasets.

Data	Method	Network construction (or clustering)	Latent representation Z	Inference	Total
g241c	AGR	0.20	1.19	0.01	1.40
	MISSL	0.20	0.08	0.01	**0.29**
	GSSL	0.25	—	0.06	0.31
g241n	AGR	0.21	1.19	0.01	1.41
	MISSL	0.21	0.08	0.01	**0.30**
	GSSL	0.25	—	0.06	0.31
MNIST	AGR	25.00	67.29	0.08	92.37
	MISSL	25.00	0.09	0.07	**25.16**
	GSSL	482.48	—	175.71	658.19
CIFAR-10	AGR	85.12	77.70	0.02	162.84
	MISSL	85.12	0.06	0.01	**85.19**
	GSSL	1,221.03	—	111.54	1,332.57

The bold values refer to the values of the proposed method (MISSL).



Suspect OOO, 20OO.OO.OO. around 03:55, the victim OOO was dragged to a warehouse located at her house and imprisoned for about 6 hours because she was drunk and noisy at her house located at OOO. The suspect pulled the victim's hair and ears and hit the victim's head.

Suspect OOO was arrested on 20OO.OO.OO. around 09:43, the victim was confined in the park bathroom for about an hour because the victim damaged suspect's belongings at the OO Park located across from OOO. The suspect assaulted the victim by pushing the victim's head and body several times.

Suspect OOO on March OO, 19OO. was sentenced to 5 months in prison and a fine of 2 million won for assault and theft of his father. The suspect is the stepfather of victim OOO. The suspect on 20OO.OO.OO. around 09:45, at the house of OOO, the victim was locked in a warehouse and confined for about 4 hours because the victim stayed outside over night.

Suspect OOO on 20OO.OO.OO. around 18:12, there was an argument at the house of victim OOO, located at OOO. The suspect locked the door when the victim entered the bathroom and went out for about 12 hours. because the victim's attitude made him angry.

Suspect OOO 20OO.OO.OO. around 05:27, at a bar located at OOO, the victim OOO was forcibly locked in the bathroom and imprisoned for about an hour because of her gossip.

Figure 4: The crime case network of 20,500 cases. The node color represents the type of crime. Less than 1% of types are grouped by "etc." The size of a node indicates the node degree. The subnetwork of "assault" is depicted on the right insert. The number in the case node represents the case report number. The cases extracted show a similar criminal pattern in which the suspects either beat or detain the victims.

whereas AGR or MISSL significantly reduces computation time by using clustering. Compared to AGR, MISSL is inherently superior to AGR in terms of speed as it does not require an optimization process. A comparison of the overall time empirically proves that MISSL is more efficient. For the small datasets, such as g241c and g241n, MISSL and AGR are 100 times and 90 times faster than the reference time, respectively, and for the large datasets, MNIST and CIFAR-10 are 1,000 times and 900 times faster.

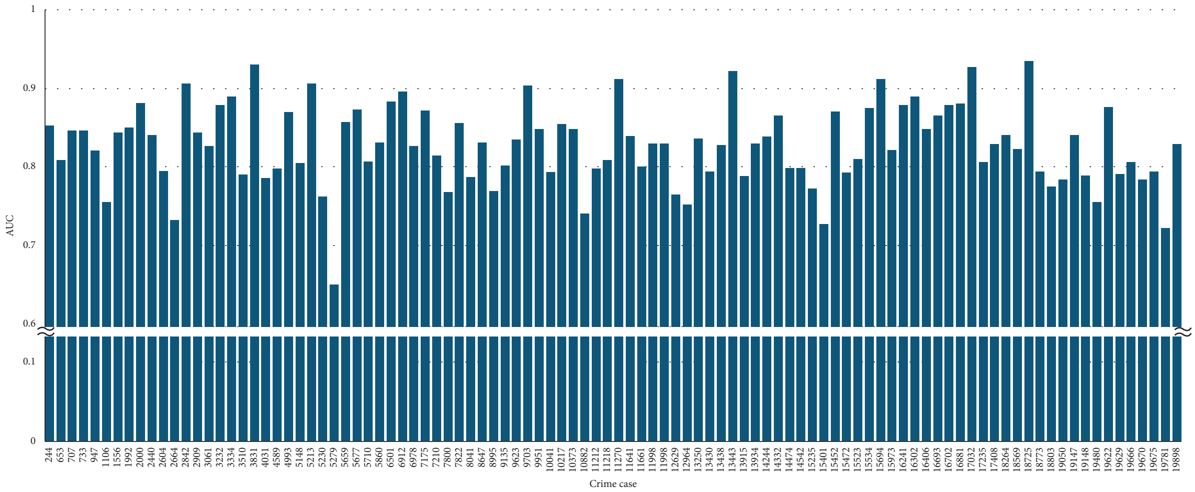
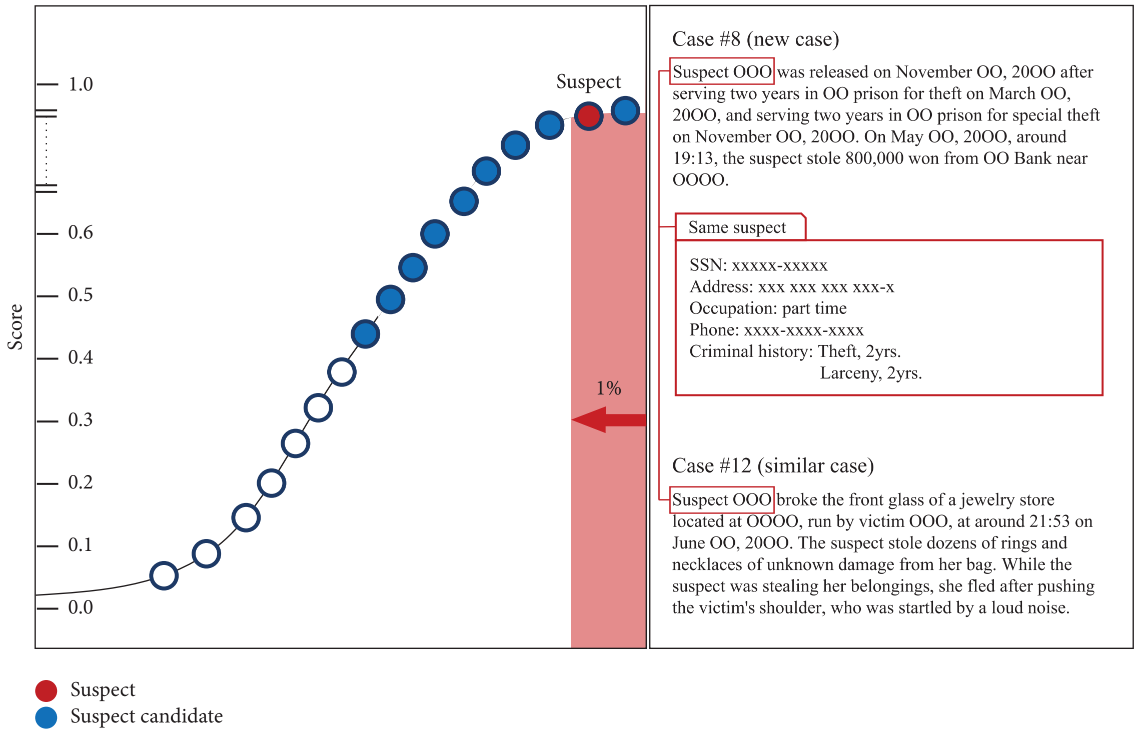


Figure 5: AUC of suspect candidate scoring for each crime case. 100 cases are randomly sampled from 20,500 cases. The worst case shows around 0.65 AUC and the best case shows around 0.93 AUC.



Case #8 (new case)

Suspect OOO was released on November OO, 20OO after serving two years in OO prison for theft on March OO, 20OO, and serving two years in OO prison for special theft on November OO, 20OO. On May OO, 20OO, around 19:13, the suspect stole 800,000 won from OO Bank near OOOO.

SSN: xxxxx-xxxxx
Address: xxx xxx xxx xxx-x
Occupation: part time
Phone: xxxx-xxxx-xxxx
Criminal history: Theft, 2yrs.
Larceny, 2yrs.

Case #12 (similar case)

Suspect OOO broke the front glass of a jewelry store located at OOOO, run by victim OOO, at around 21:53 on June OO, 20OO. The suspect stole dozens of rings and necklaces of unknown damage from her bag. While the suspect was stealing her belongings, she fled after pushing the victim's shoulder, who was startled by a loud noise.

Figure 6: Scores of suspect candidates for case #8. Blue circles indicate suspect candidates, and a red circle indicates actual suspects. The suspect is in the top 1% of suspect candidates. Case #12 was similar to case #8.

5. Applications for Suspect Candidate Scoring Using the Criminal Network

We applied MISSL to real crime data provided by the Korean National Police. The experimental setup and results are as follows.

5.1. Criminal Network and Experimental Setting. Figure 4 visualizes clusters in the case network using OpenOrd [46]. The node color represents the type of crime. The same type of crime is grouped closely in the network and batteries and assaults are grouped together since they are both violent crimes. For validation on suspect scoring, a case node is randomly selected as a test node, and the edges connected to the node are removed from the network (including the edge to the actual suspect). The top 20 people with the highest scores were reported as suspect candidates. The performance is measured by calculating the AUC of the predicted suspects for each crime case. The leave-one-out (LOO) method is applied to 20,500 cases.

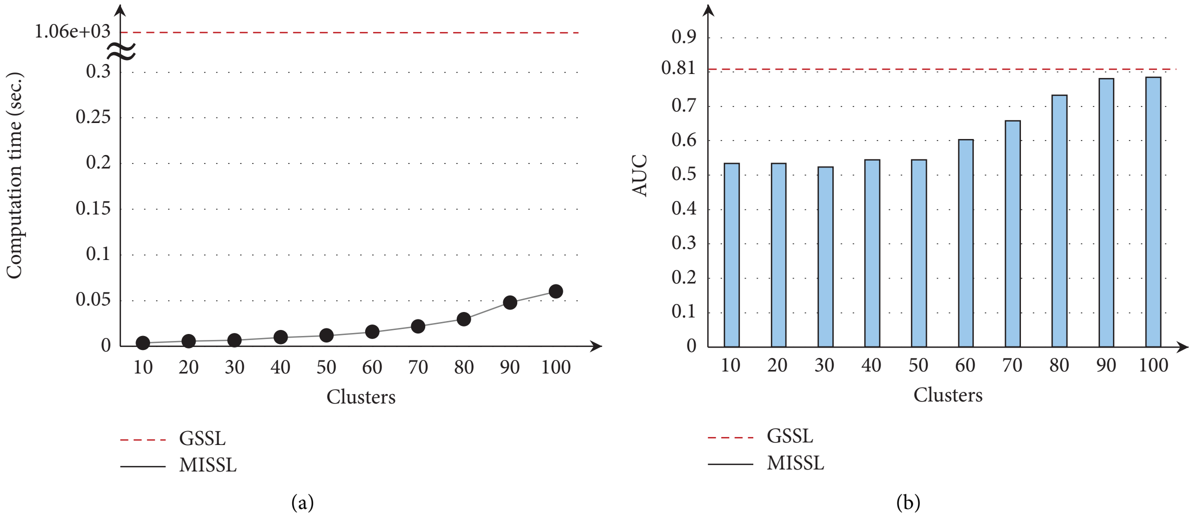


FIGURE 7: The computation time (line chart) and AUC performance (bar chart) of MISSL while varying the number of clusters. The performance of GSSL is shown by red dotted lines.

5.2. Results of Suspect Candidate Scoring. In Figure 5, the AUCs for 100 cases are shown. The overall average AUC for 20,500 cases is 0.82 ± 0.01. More specifically, Figure 6 presents a typical scoring curve. Suspect candidates for case #8 are sorted along the curve. The blue nodes represent predicted suspect candidates and the red represents the actual suspect. As shown in figure, the real suspect is ranked relatively high, within the top 1%. Case #12 was similar to case #8. Both cases are similar in that the suspect steals money or jewelry. The summary reports of cases #8 and #12 are described on the right side with the personal information of the suspect. Indeed, the suspect of case #12 was highly scored (red circle) and was the actual offender of case #8.

Figure 7 shows the computation time and performance of MISSL while varying the number of clusters or centroids, respectively. GSSL are indicated by red and blue dotted lines in the figure. By comparing the computation time, it was found that the MISSL was 0.06 seconds when the number of clusters was 100, which is about $1.76e + 04$ times faster than that of GSSL (1,058.65 seconds). The computation time increases as the number of clusters increases but is still very trivial compared to GSSL. The performance of MISSL increases as the number of clusters increases. The highest AUC of 0.8 was reached when the number of clusters was 100. From the perspectives of both computational time and performance, MISSL provides a high accuracy within a reasonable amount of time. This is critical in real-world applications. In an urgent case, the police do not have to wait all day to get a list of suspect candidates.

6. Conclusion and Discussion

In this study, we proposed a framework for predicting suspect candidates based on the criminal network. The algorithm we employed is graph-based SSL, which may be inappropriate when networks are large and complicatedly structured. So, to put the GSSL to practical use in the criminal network, we developed an algorithm based on latent representation and mutual information. The proposed method, MISSL, shows almost similar performance to the graph-based SSL but has a much faster inference time. As an application, a criminal network is constructed from real-world crime data, and suspect candidate scoring is performed by MISSL. The predicted results show the validity and efficacy of MISSL. The framework of suspect candidate scoring introduces a novel way of analyzing crime data, and the results shed a new light on network-based machine learning approaches for social network analysis. Future efforts may identify a more efficient mechanism to optimize the hyperparameter of MISSL and the number of clusters (i.e., the number of latent dimensions) that affect performance. Empirically, a higher AUC was obtained from the increased number of clusters. From another perspective, clustering can be expensive for large-scale datasets. Therefore, further research on faster clustering methods or modeling the algorithms robust to clustering will enrich the results of our study.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Institute for Information Communications Technology Promotion (IITP) grant funded by the Korea Government (MSIT) ((Grant no. 2022-0-00653) Voice Phishing Information Collection and Processing and Development of a Big Data-Based Investigation Support System), BK21 FOUR program of the National

Research Foundation of Korea funded by the Ministry of Education (Grant no. NRF5199991014091), and the Ajou University research fund.

References

[1] F. U. M. Ullah, M. S. Obaidat, K. Muhammad et al., “An intelligent system for complex violence pattern analysis and detection,” *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 10400–10422, 2022.

[2] D. Sardana, S. Marwaha, and R. Bhatnagar, “Supervised and unsupervised machine learning methodologies for crime pattern analysis,” *International Journal of Artificial Intelligence and Applications*, vol. 12, no. 1, pp. 83–99, 2021.

[3] Z. Yan, H. Chen, X. Dong, K. Zhou, and Z. Xu, “Research on prediction of multi-class theft crimes by an optimized decomposition and fusion method based on XGBoost,” *Expert Systems with Applications*, vol. 207, Article ID 117943, 2022.

[4] S. Prabakaran and S. Mitra, “Survey of analysis of crime detection techniques using data mining and machine learning,” *Journal of Physics: Conference Series*, vol. 1000, no. 1, Article ID 012046, 2018.

[5] L. McClendon and N. Meghanathan, “Using machine learning algorithms to analyze crime data,” *Machine Learning and Applications: International Journal*, vol. 2, no. 1, pp. 1–12, 2015.

[6] S. Vijayarani, E. Suganya, and C. Navya, “A comprehensive analysis of crime analysis using data mining techniques,” *International Journal of Computer Science and Engineering*, vol. 9, no. 1, 2020.

[7] M. K. Sparrow, “The application of network analysis to criminal intelligence: an assessment of the prospects,” *Social Networks*, vol. 13, no. 3, pp. 251–274, 1991.

[8] D. Bright, R. Brewer, and C. Morselli, “Using social network analysis to study crime: navigating the challenges of criminal justice records,” *Social Networks*, vol. 66, pp. 50–64, 2021.

[9] K. Faust and G. E. Tita, “Social networks and crime: pitfalls and promises for advancing the field,” *Annual Review of Criminology*, vol. 2, no. 1, pp. 99–122, 2019.

[10] M. Weber, J. Chen, T. Suzumura et al., “Scalable graph learning for anti-money laundering: a first look,” 2018, https://arxiv.org/abs/1812.00076.

[11] P. Das, A. K. Das, J. Nayak, D. Pelusi, and W. Ding, “A graph based clustering approach for relation extraction from crime data,” *IEEE Access*, vol. 7, pp. 101269–101282, 2019.

[12] P. Bangcharoensap, H. Kobayashi, N. Shimizu, S. Yamauchi, and T. Murata, “Two step graph-based semi-supervised learning for online auction fraud detection,” in *Joint European conference on machine learning and knowledge discovery in databases*, Springer, Cham, Switzerland, August 2015.

[13] H. Alvari, P. Shakarian, and J. E. Snyder, “Semi-supervised learning for detecting human trafficking,” *Security Informatics*, vol. 6, no. 1, pp. 1–14, 2017.

[14] M. Khan, A. Ali, and Y. Alharbi, “Predicting and preventing crime: a crime prediction model using san francisco crime data by classification techniques,” *Complexity*, vol. 2022, Article ID 4830411, 2022.

[15] D. D. Lopes, B. R. Cunha, A. F. Martins et al., “Machine learning partners in criminal networks,” *Scientific Reports*, vol. 12, no. 1, Article ID 15746, 2022.

[16] G. Berlusconi, F. Calderoni, N. Parolini, M. Verani, and C. Piccardi, “Link prediction in criminal networks: a tool for criminal intelligence analysis,” *PLoS One*, vol. 11, Article ID e0154244, 2016.

[17] F. Calderoni, S. Catanese, P. De Meo, A. Ficara, and G. Fiumara, “Robust link prediction in criminal networks: a case study of the Sicilian Mafia,” *Expert Systems with Applications*, vol. 161, Article ID 113666, 2020.

[18] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, “Learning with local and global consistency,” *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[19] D. Zhou and B. Schölkopf, “A regularization framework for learning from graph data,” in *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields (SRL 2004)*, Banff, Canada, December 2004.

[20] F. Hoffmann, “Consistency of semi-supervised learning algorithms on graphs: probit and one-hot methods,” *Journal of Machine Learning Research*, vol. 21, pp. 1–55, 2020.

[21] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: a geometric framework for learning from labeled and unlabeled examples,” *Journal of Machine Learning Research*, vol. 7, p. 11, 2006.

[22] Z. Xu, I. King, M. R. T. Lyu, and J. Rong, “Discriminative semi-supervised feature selection via manifold regularization,” *IEEE Transactions on Neural Networks*, vol. 21, no. 7, pp. 1033–1047, 2010.

[23] D. Zhou, J. Huang, and B. Schölkopf, “Learning from labeled and unlabeled data on a directed graph,” in *Proceedings of the 22nd international conference on Machine learning*, Bonn, Germany, August 2005.

[24] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using Gaussian fields and harmonic functions,” in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, Washington, DC, USA, January 2003.

[25] T.-P. Nguyen and T. B. Ho, “Detecting disease genes based on semi-supervised learning and protein–protein interaction networks,” *Artificial Intelligence in Medicine*, vol. 54, no. 1, pp. 63–71, 2012.

[26] H. Shin, K. Tsuda, and B. Schölkopf, “Protein functional class prediction with a combined graph,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 3284–3292, 2009.

[27] D.-G. Lee, S. Lee, M. Kim, and H. Shin, “Historical inference based on semi-supervised learning,” *Expert Systems with Applications*, vol. 106, pp. 121–131, 2018.

[28] M. Kim, D. Lee, S. Lee, G. Lee, and H. Shin, “Inference on historical factions based on multi-layered network of historical figures,” *Expert Systems with Applications*, vol. 161, Article ID 113703, 2020.

[29] H. Huang, Y. Cao, X. Huang, H. Ji, and C. Y. Lin, “Collective tweet wikification based on semi-supervised graph regularization,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 1, Long Papers, Baltimore, MD, USA, June 2014.

[30] M. Balaanand, N. Karthikeyan, S. Karthik, R. Varatharajan, G. Manogaran, and C. B. Sivaparthipan, “An enhanced graph-based semi-supervised learning algorithm to detect fake users on Twitter,” *The Journal of Supercomputing*, vol. 75, no. 9, pp. 6085–6105, 2019.

[31] Y. Shao, N. Sang, C. Gao, and L. Ma, “Spatial and class structure regularized sparse representation graph for semi-supervised hyperspectral image classification,” *Pattern Recognition*, vol. 81, pp. 81–94, 2018.

[32] A. Subramanya, S. Petrov, and F. Pereira, “Efficient graph-based semi-supervised learning of structured tagging models,” in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Massachusetts, MA, USA, October 2010.

[33] M. Sanderson, P. Raghavan, and H. Schütze, "Christopher D. Manning, prabhakar raghavan, hinrich schütze, introduction to information retrieval, cambridge university press. 2008. ISBN-13 978-0-521-86571-5, xxi + 482 pages," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.

[34] KICS, "Korea information system of criminal justice Services (KICS)," 2022, https://www.kics.go.kr/.

[35] E. L. Park and S. Cho, "KoNLPy: Korean natural language processing in Python," in *Annual Conference on Human and Language Technology*, Banff, Canada, January 2014.

[36] M. Kim, D.-G. Lee, and H. Shin, "Semi-supervised learning for hierarchically structured networks," *Pattern Recognition*, vol. 95, pp. 191–200, 2019.

[37] J. H. Jhee, M. J. Kim, M. Park, J. Yeon, and Y. Kwak, "Fast prediction for suspect candidates from criminal networks," in *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, IEEE, Jeju, Korea, February 2023.

[38] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review*, vol. 69, no. 6, Article ID 066138, 2004.

[39] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[40] D. Zhou, T. Hofmann, and B. Schölkopf, "Semi-supervised learning on directed graphs," *Advances in Neural Information Processing Systems*, vol. 17, 2004.

[41] O. Chapelle, B. Scholkopf, and A. Zien Eds, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, p. 542, 2009.

[42] The Mnist database, "MNIST handwritten digit database, yann LeCun, corinna cortes and chris burges," 2022, http://yann.lecun.com/exdb/mnist/.

[43] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009, http://www.cs.utoronto.ca/%7Ekriz/learning-features-2009-TR.pdf.

[44] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, New York, NY, USA, June 2010.

[45] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[46] S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack, "OpenOrd: an open-source toolbox for large graph layout," *Visualization and Data Analysis 2011*, Vol. 7868, International Society for Optics and Photonics, Washington, DC, USA, 2011.