ISSN 2383-6318(Print) / ISSN 2383-6326(Online)
KIISE Transactions on Computing Practices, Vol. 25, No. 9, pp. 436-445, 2019. 9
https://doi.org/10.5626/KTCP.2019.25.9.436

급성 신장 손상 예측을 위한 의료 데이터 전처리
(Medical Data Preprocessing For Predicting Acute Kidney Injury)

양 현 식 † 임 유 빈 † 이 진 영 ††
(Hyeonsik Yang) (Yubin Lim) (Jinyeong Yi)

김 동 효 ††† 김 세 중 †††† 김 형 주 †††††
(Donghyo Kim) (Sejoong Kim) (Hyoungjoo Kim)

요 약 의료 정보의 활용성이 증대되면서 의료 영역에 데이터 분석 기법을 접목하려는 노력들이 많이 시도되고 있지만, 실제 의료 데이터는 다양한 분과, 담당 의사, 병동으로 세분화되어 비표준화·파편화되어 있기 때문에 분석에 활용하기 어려운 경우가 많다. 이러한 특성을 고려하여 분석에 적합한 형태로 정제하는 전처리 작업이 필수적으로 선행되어야 하지만, 실제적인 전처리 작업에 대한 연구는 거의 이루어지지 않고 있는 상황이다. 본 논문에서는 급성 신손상(AKI, 급성 신장 손상) 예측이라는 구체적인 사례를 기반으로 의료 데이터의 특성을 반영하여 데이터를 분석에 적합한 형태로 정제하는 전처리 과정을 설계하고 상세히 서술한다. 작업 형태 별로 데이터 클리닝, 데이터 통합, 데이터 변환, 데이터 축소, 데이터 이산화로 구분한 전처리 작업들을 활용하여 데이터를 분석에 적합한 형태로 정제하고, 간단한 실험을 통해 정제된 데이터가 유효하고 효과적으로 작용함을 확인한다.

키워드: 전처리, 의료 데이터 분석, 발병 예측, 급성 신손상

Abstract As the utilization of medical information increases, many attempts have been made to incorporate data analysis techniques into the medical field. However, since actual medical data, divided between various specialties, doctors, and wards, is non-standardized and fragmented, it is often difficult to utilize it. Because of this, it is essential to perform preprocessing steps to refine the data into a form suitable for analysis. However, there is little research on practical preprocessing. In this paper, we designed and detailed a preprocessing process to refine data into a form suitable for analysis. The process was based on the specific use of data to predict acute kidney injury and reflected the characteristics of the medical data. Using various preprocessing tasks, classified into data cleaning, data integration, data transformation, data reduction, and data discretization according to the work type, we refined the data and confirmed its validity and effectiveness.

Keywords: data preprocessing, medical data analysis, disease prediction, acute kidney injury

· 이 논문은 2019년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.R0113-15-0005, 대규모 트랜잭션 처리와 실시간 복합 분석을 통합한 일체형 데이터 엔지니어링 기술 개발)

† 비 회 원 : 서울대학교 컴퓨터공학부(Seoul Nat'l Univ.)
hsyang@idb.snu.ac.kr
yblim@idb.snu.ac.kr
(Corresponding author임)
†† 비 회 원 : 분당 서울대 병원 신장내과 연구원
julykid@hanmail.net
††† 비 회 원 : 서울대학교 컴퓨터공학부
dhkim@idb.snu.ac.kr
†††† 비 회 원 : 분당 서울대 병원 신장내과 교수
sejoong2@snu.ac.kr
††††† 종신회원 : 서울대학교 컴퓨터공학부 교수
hjk@snu.ac.kr

논문접수 : 2019년 4월 9일
(Received 9 April 2019)
논문수정 : 2019년 7월 1일
(Revised 1 July 2019)
심사완료 : 2019년 7월 8일
(Accepted 8 July 2019)

정보과학회 컴퓨팅의 실제 논문지 제25권 제9호(2019. 9)

1. 서 론

의료기관 내 의료 정보의 전산화 비율이 높아지고 분석 기법 및 장비와 같은 정보화 기술의 발전으로 의료 정보의 활용성이 증대되면서, 의료 영역에 데이터 분석 기법을 접목시켜 기존에 해결하기 어려웠던 문제들을 풀고자 하는 연구들이 많이 진행되어왔다.

하지만, 실제 의료 데이터는 일관된 기준 없이 다양한 분과, 담당 의사, 병동 별로 독립적으로 기록되기 때문에 표준화되어 있지 않고 파편화되어있다. 정돈되지 않은 데이터는 대부분 결함을 가지거나 및 비일관성 문제를 포함하여 분석에 바로 활용할 수 없는 경우가 많으므로, 분석에 적합한 형태로 정제하는 전처리 작업을 필요로 한다. 전처리 과정은 분석 모델의 성능에도 직접적인 영향을 미치는 중요한 과정이므로, 데이터 분석에 앞서 효과적인 전처리 작업의 선행은 필수적이다. 하지만, 많은 연구에서 전처리 작업의 중요성을 강조하고 있으나, 전처리 방법에 대해 구체적으로 기술하거나 그 효과에 대해 입증하는 연구는 부족한 상황이다.

본 논문에서는 실제 의료 데이터를 활용하여 분석 모델을 학습하고 특정 진단을 내려야 하는 상황을 상정하고, 올바른 모델을 학습하기 위해 필요한 데이터 전처리 방법에 대한 연구를 진행하였다. 본 연구에서는 급성 신장 손상 영역의 데이터를 활용하여 급성 신장 손상을 예측하는 상황을 상정하고, 다섯 가지 주요 전처리 작업을 활용하여 데이터를 정제하는 과정을 구체적으로 기술한다. 전처리 작업을 수행한 후에 단순한 기계 학습 모델을 이용한 간단한 실험을 통해 전처리 작업의 적합성을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 급성 신장 손상 질병의 예측 문제의 중요성에 대해 소개하며, 기존의 급성 신장 손상 예측 시스템과 기계 학습 알고리즘을 사용한 예측 모델의 한계점을 다루고, 주요 전처리 작업에 대한 소개 및 관련 연구를 소개한다. 3장에서는 다섯 가지 주요 전처리 작업에 기반하여 실제 수집한 급성 신장 손상 의료 데이터를 정제하는 과정을 상세히 다룬다. 4장에서는 정제된 데이터를 사용하여 기계 학습 모델을 학습시키고, 모델의 성능을 평가하여 전처리 작업의 적합성을 확인한다. 마지막으로 5장에서 결론과 향후 연구에 관해 기술한다.

2. 관련연구

2.1 급성 신장 손상

급성 신장 손상이란 일반적으로 7일 이내에 발생하는 갑작스런 신장 기능의 상실이다. 구체적으로는 혈청 크레아티닌의 수치가 48시간 이내에 0.3mg/dL이상 상승하거나 이전 일주일 내의 수치보다 1.5배 이상 상승하는 경우를 급성 신장 손상으로 정의한다. 급성 신장 손상은 다양한 원인에 의해 발생할 수 있으므로 원인 질환에 따른 치료법 및 예후법 또한 다양하다. 급성 신장 손상은 이환율과 사망률이 높고, 경증의 손상에도 사망률을 증가시키는 경과를 보일 수 있으므로 조기에 발견해서 치료를 빠르게 시작하는 것이 중요하다. 조기 발견을 놓쳐서 급성 신장 손상이 지속된 경우는 환자의 수분, 전해질 및 산-염기 불균형과 호르몬 조절 능력의 소실을 유발하며 다장기 부전을 유발할 수 있다. 반면, 급성 신장 손상의 조기 발견을 통한 적절한 치료는 환자의 생존율을 향상시키는데 유의미한 상관 관계를 보인다[1]. 최근에는 예측을 통한 사전 대처로 급성 신장 손상의 위험성을 최소화하기 위한 연구들이 진행되어 왔고, 실제 병원에서도 활용되고 있다.

BENECIA[1]는 2014년부터 분당 서울대병원에서 시행중인 급성 신장 손상 감시 프로그램으로, 원내에서 발생한 급성 신장 손상을 전산 시스템을 통하여 발견하고 이를 통해 신속한 치료를 제공하기 위한 목적으로 시행되었다. 하지만, 단순히 감시 용도로 설계되었기 때문에 예측같이 데이터를 활용하는 것은 불가능하다. 이와 달리, SPARK[2]는 수술 전 환자 정보, 검사 결과, 예상 수술 시간 등을 기반으로 환자의 수술 후 2주 이내 급성 신장 손상 발생 여부를 예측하는 모델이다. 하지만, 수술 예정이 없는 일반 환자들에 대해서는 적용이 불가능하다는 한계를 가진다. 이 외에도, 기계 학습 기법을 활용하여 급성 신장 손상을 예측하고자 하는 연구들이 꾸준히 진행되고 있다[2-4]. [2]는 Logistic regression, SVM 등 4가지 기계 학습 알고리즘을 활용하여 급성 신장 손상을 예측하는 모델을 제안하였지만, 모델 학습에 사용된 데이터의 표본이 60세 이상의 환자들로 제한되어 있다는 한계를 가진다. [3]에서도 약 20여 가지의 기계 학습 알고리즘을 활용해 급성 신장 손상을 예측하는 연구를 수행하였지만, 표본이 심장 수술을 받은 경력이 있는 환자들로 한정된다는 한계를 가진다. [4]은 일반적인 환자들에 대해 급성 신장 손상을 예측하여 좋은 성능을 보였지만, 실험을 소급 적용하여 수행하였다는 한계가 존재한다.

이처럼 급성 신장 손상을 예측하기 위해 다양한 연구들이 진행되어 왔지만, 데이터의 특성을 고려한 전처리 과정은 생략되거나 간략하게 언급되어 있어 보다 구체적인 사례를 통한 데이터 처리 방법의 제시가 필요하다.

---

1) Benefit of early detection and nephrologic intervention in critical ill patients
2) Simple Postoperative-AKI Risk classification in major non-cardiovascular surgery

2.2 데이터 전처리

실제 데이터 분석을 위해 수집한 대부분의 데이터는 다양한 문제점을 내포하고 있어 데이터 분석에 활용하기 어렵다. 문제가 되는 데이터들은 크게 불완전(Incomplete), 노이즈(Noisy), 비일관(Inconsistent) 데이터 세 가지로 분류 가능하다. 먼저, 불완전 데이터는 데이터 중 필요한 속성이나 정보가 없는 경우, 노이즈 데이터는 데이터 중에 정확하지 않은 값이나 정상 범위 밖의 이상치(Outlier)가 있는 경우를 의미한다. 마지막으로, 비일관 데이터는 여러 데이터에 동일한 의미를 가지는 정보가 다르게 기록되어 있는 경우로, 데이터들을 합칠 때 문제가 발생하게 된다.

그러므로, 데이터 분석에 앞서 데이터들을 정제하는 전처리 과정이 필수적으로 선행되어야 한다. 전처리는 데이터 분석 결과에 직접적인 영향을 미치므로, 많은 시간을 할애해서라도 데이터에 적합한 전처리 작업을 적용하는 것은 중요하다. [5]은 전처리 작업을 크게 데이터 클리닝, 데이터 통합, 데이터 변환, 데이터 축소, 데이터 이산화의 다섯 가지로 구분하였다. 각 작업 별 구체적인 작업 내용은 표 1에 정리하였다.

수집한 데이터를 전처리하여 유의미한 분석 결과를 도출하기 위한 연구는 다양한 분야에서 진행되어 왔다[6,7]. [6]에서는 웹 사이트 이용 패턴 분석을 위한 로그 파일 전처리 방법론을 제시하였고, [7]에서는 맵리듀스 프레임워크를 이용하여 영화 흥행 실적 예측을 위한 전반적인 데이터 전처리 과정을 소개하였다.

마찬가지로, 의료 분야에서도 전처리 작업을 고려한 연구가 진행되어 왔다[8-11]. [8]는 다양한 의료 데이터에 대한 분류 실험을 통해 특징 선택이나 인스턴스 선택 같은 데이터 축소 작업과 K-Nearest Neighbors 기반 결측치 대체 작업이 분석 성능에 주는 영향을 확인하였고, [9]은 다양한 전처리 방법을 올바르게 조합하는 것이 분석 성능의 향상으로 이어질 수 있음을 보였다. [10]은 전처리를 통해 데이터의 불균형, 다차원성 문제를 해결한 후 골다공증 예측 모델을 학습하였고, [11]은 노쇠 증후군 예측을 위한 결측치 대체, 특징 선택, 정규화 방법을 자세히 소개하였다. 이 외에도, 의료 데이터를 활용하기 위한 다양한 연구들이 진행되고 있고[2-4], 전처리 작업의 중요성도 강조하고 있다. 하지만, 실제 병원 데이터보다는 분석을 위해 정돈된 데이터를 사용하는 경우가 많아, 활용하는 전처리 작업도 일부로 한정되어 있거나 작업 과정에 대한 구체적인 설명이 생략되어 있는 경우가 많다. 또한, 의료 데이터는 질병이나 예측 목적에 따라 필요로 하는 데이터의 형태가 달라지기 때문에 데이터를 정확하게 파악하는 것이 중요하지만, 이에 대한 언급도 부족한 경우가 많다. 따라서, 의료 데이터를 보다 실제적으로 분석에 활용하기 위해서는 구체적인 사례를 통해 전반적인 전처리 작업의 예시가 필요하다.

표 1 데이터 전처리의 다섯 가지 주요 작업
Table 1 5 Major data preprocessing tasks

Data Cleaning	- Impute missing value - smooth noisy data - Identify or remove outliers - solve data inconsistency
Data Integration	- Integrate multiple databases, files
Data Transformation	- Data normalization - Data Summarization - Data aggregation
Data Reduction	- Feature Selection - Instance Selection
Data Discretization	- Convert numerical values to categorical values

3. 의료 데이터 전처리

3.1 원본 데이터

본 절에서는 전처리 측면에서 급성 신장 손상 예측을 위해 수집한 분당 서울대병원의 환자 데이터를 분석하고 이에 대한 처리 방법을 제시한다.

본 논문에서는 2013~2017년 동안 분당 서울대병원에 입원했던 102,721명의 환자를 대상으로 기록한 178,430건의 비공개 데이터를 사용한다. 해당 데이터는 다양한 병동/분과/담당의사 등을 기준으로 나뉘어 있는 364개의 비표준화, 파편화된 엑셀 파일로 존재한다. 즉, 일관된 제약 사항이나 기준 없이, 데이터를 기록한 주체 별로 서로 다른 기준에 따라 기록된 정리되지 않은 데이터이다. 예를 들어, 누락되거나 잘못된 정보가 기록되어 있거나 파일 별로 이름이 동일하지만 의미가 다른 컬럼이 존재하기도 하며 그 반대의 경우도 존재한다. 원본 데이터의 요약은 표 2에 정리하였다.

본 절의 나머지 부분에서는 표 2와 같이 수집된 데이터를 분석하여 데이터의 특성을 파악하고, [5,12]에 정리된 다섯 가지 데이터 전처리 작업을 활용하여 데이터를 분석에 활용할 수 있는 형태로 정제하는 과정을 상세히 기술한다.

3.2 원본 데이터 1차 통합

364개 파일로 구성된 원본 데이터를 하나씩 분석하여 전처리를 수행하는 것은 비효율적이므로, 파일의 수를 줄이기 위해 각 파일의 스키마 분석을 먼저 수행하였다. 스키마 분석을 통해 수많은 파일들 중 일련의 파일들은 유사한 스키마를 가진다는 것을 확인하였는데, 이는 양질의 의료 서비스 제공을 위해 분과나 검사 내용 등을

표 2 원본 데이터 요약
Table 2 Brief description of raw data

파일명	컬럼명
Adm. & D/C	ID, Dept., DOA(Date of Admission), DOD(Date of Discharge)
Adm. Info.	ID, adm_date, Bldg.
Dialysis	ID, Rx_date, Rx
Kidney Tx.	ID, OP_date, OP
Anesthesia	ID, OP_date, OP, Anes_type
Dianosis	ID, Dx_date, Dx, Dx_Code
Ammonia.t_bil	ID, trmt_date, Ex_date, Ex, Ammonia, Bilirubin
Ca.glucose	ID, trmt_date, Ex_date, Ex, Calcium, Glucose
Examination	ID, trmt_date, Ex_date, Ex, BUN, O2SAT, O2CT, BE, PLT, WBC, HCO3
Colistin	ID, Rx_date, Med_Code, Med
Cyclosporine	ID, Rx_date, Med_Code, Med
Aminoglycoside	ID, Rx_date, Med_Code, Med

세분화한 실제 의료 시스템의 특성에 기인한다. 예를 들어, 표 2의 “Ammonia.t_bil”, “Ca.glucose”, “Examination” 3개 파일은 스키마가 유사한데, 이 파일들이 모두 병원에서 수행한 검사 결과 정보를 기록한 파일이기 때문이다. “Ammonia.t_bil”에는 Ammonia와 Bilirubin과 관련된 검사 내역들만 기록하고, “Ca.glucose”에는 Calcium과 Glucose와 관련된 검사 내역들만 기록한 것이다. 즉, 모든 검사 내역을 통합하여 기록하지 않고, 중요한 요소 별로 세분화하여 기록한다. 유사하게 “Colistin”, “Cyclosporine”, “Aminoglycoside” 3개 파일의 경우는 처방약의 종류에 따라서 처방 내역을 세분화하여 기록한 것이다. Colistin, Aminoglycoside는 항생 물질의 일종, Cyclosporine은 거부 반응 방지제의 일종이다. 따라서, 데이터를 “검사”나 “처방”같은 보다 일반적인 분류 기준으로 통합하는 것이 가능하다.

위에서 파악한 원본 데이터의 특성과 전문가 의견을 기반으로 총 5가지의 기준을 선정하여 데이터를 1차적으로 통합하였다(표 3). 통합 기준은 다음과 같다. 1)환자의 기본 정보(Patient Information) 2)검사 내역(Examination) 3)수술 내역(Operation) 4)처방 내역(Prescription) 5)진단 내역(Diagnosis). 이 중에서 환자 정보는 개인 정보 보호 문제로 인하여 암호화된 환자ID, 나이와 입원 날짜를 제외하고 모두 제거하였다. 통합 데이터는 이후 전처리 과정에서 빈번한 스키마 변경과 추가적인 통합 과정을 필요로 하므로 효율적인 작업을 위해 데이터베이스에 저장하기로 한다. 이후부터의 설명에서는 데이터베이스 사용을 가정한다.

표 3 1차 통합 테이블 스키마
Table 3 Primary integration table schema

테이블명	컬럼명
Patient Information (Info)	ID, Age, DOA
Examination (Ex)	Ex_Date, ID, Hb, Albumin, Creatinine, eGFR, Sodium, Protein, Hb, BUN, Cr, tCO2, HCO3-, PCO2, pH, Ammonia, Ca, Glucose, Na, K, Cl, CK, CK-MB, etc.
Operation (OP)	OP_Date, ID, OP, OP_time, Anes., transfusion
Prescription (Rx)	Rx_Date, ID, RAAS Blocker, colistin, aminoglycoside, cyclosporine, etc.
Diagnosis (Dx)	Dx_Date, ID, Dx, Dx_code

3.3 노이즈 처리를 위한 데이터 클리닝

통합 데이터에는 환자 기본 정보와 병원 의료 시스템에서 입력되어 노이즈가 전혀 없는 컬럼도 존재하지만, 병원에서 환자를 치료하는 과정에서 편의를 위해 생성한 컬럼도 존재한다. 이 컬럼의 데이터는 병원 운영과 환자 관리의 편의를 위해 존재하는 것이기 때문에 분석에는 불필요할 뿐만 아니라 분석 모델의 성능을 감소시킨다. 따라서 효과적이고 효율적인 의료 데이터 분석을 위해 환자의 상태와 치료에 관련된 컬럼 외에는 모두 삭제해야 한다. 레코드 자체가 부적절한 경우도 있다. 예를 들어, “부적합”, “미평가”, “신뢰할 수 없음”과 같이 의미 없는 레코드가 존재하기도 하고, 데이터 수집 당시의 문제로 “>40”, “60(80)”또는 “+”, “-”와 같이 분석 모델이 인식하지 못하는 형식으로 기록된 레코드도 존재한다. 올바른 분석을 위해서는 이러한 데이터들은 삭제하거나 모델이 인식할 수 있는 형태로 변환하는 데이터 클리닝 작업이 필요하다. 이 작업을 위한 데이터 클리닝 알고리즘은 그림 1과 같다.

제안하는 데이터 클리닝 알고리즘은 하나의 테이블을 입력받아 클리닝을 진행한다. 먼저, 테이블의 각 레코드에서 컬럼을 추출하여 해당 컬럼이 숫자를 포함하는지 확인한다(2-4). 컬럼이 숫자를 포함할 때 가능한 형태는 “초기 검사 값(재검사 값)” 형태, “>40”처럼 기호와 같이 사용된 형태 혹은 “20.7”처럼 완전한 실수만 존재하는 3개의 경우로 나뉜다. 먼저, “초기 검사 값(재검사 값)” 형태의 데이터는 재검사 값만을 이용하기로 한다. 이를 위해 컬럼의 문자열을 ‘(’문자로 문자열을 분할하여 0번째 인덱스의 “초기 검사 값”은 버리고 1번째 인덱스의 “재검사 값)”을 사용한다(5-6). 이어서, line 7에서 숫자 외의 정보들을 모두 제거하여 수치화하는 것으로 숫자를 포함한 컬럼의 클리닝은 완료된다. 그 외에 컬럼에 숫자가 없이 ‘-’만으로 구성되어 있으면 0, ‘+’만으로

Algorithm 1 Data Cleaning

```
1: Method Cleaning(table)
2:   FOR EACH record IN table:
3:     FOR EACH column IN record:
4:       IF column.hasDigit() :
5:         IF column.has("(") :
6:           column ← column.split(" (")[1]
7:         column ← RemoveNonnumerics(column)
8:       ELSE IF column.has("-") :
9:         column ← 0
10:      ELSE IF column.has("+") :
11:        column ← 1
12:      ELSE
13:        column ← NULL
```

그림 1 노이즈 데이터 클리닝 과정
Fig. 1 Noisy data cleaning process

구성되어 있으면 1을 입력한다(8-11). 만약 위 경우에 해당하지 않는다면 이는 분석 시 해석하기 어려운 노이즈 데이터이므로 삭제한다(12-13).

전문가의 의견에 따라 기존에 존재하거나 위 과정 중에 발생한 null 값들은 당장 처리하지 않고 데이터 전처리 작업을 더 진행된 후에 처리하기로 결정하였다.

3.4 중복 컬럼 처리를 위한 축소 및 변환

데이터가 병동/분과/담당 의사 별로 독립적으로 수집되었기 때문에 통합된 "검사" 테이블에는 명칭은 다르지만 동일한 의미를 가지는 컬럼들이 존재한다. 예를 들어, "검사" 테이블에 존재하는 "Creatinine"과 "Cr"이라는 두 컬럼은 모두 환자의 크레아티닌의 수치를 기록한 컬럼이다. 이러한 컬럼이 "검사" 테이블 내에 다수 존재하는데, 검사 시 오차로 인해 동일한 의미를 가지는 컬럼이어도 기록된 정보가 다른 경우가 존재한다.

따라서, 임의로 하나의 컬럼을 제거하기보단 모든 컬럼의 정보를 반영하기 위하여 검사 시 평균 값을 계산한 새로운 컬럼을 생성하고 기존 컬럼들을 제거하는 방식으로 데이터의 변환과 축소 작업을 동시에 진행하였다. 크레아티닌의 경우, 급성 신장 손상 예측을 위한 baseline의 정보도 필요하므로 최소 값도 새 컬럼으로 추가하였다. 그 결과 총 18개의 요소들에 대해 변환 및 축소 작업이 진행되었다.

추가로, "검사" 테이블에는 동일한 검사가 하루에 두 번이상 시행된 각각의 경우를 모두 개별 레코드로 기록하고 있는데, 이런 레코드들은 컬럼 변환 및 축소 작업을 완료한 후에 날짜를 기준으로 하나의 레코드로 축소하였다. 축소 시에도 마찬가지로 레코드들의 각 컬럼에 평균 값을 기입하였다.

3.5 데이터 변환

3.5.1 데이터의 시계열 변환

본 연구에서 사용한 데이터에서 환자에 대한 검사 및 처방 등은 입원 후 며칠에 걸쳐 수행된다. 이런 내역들은 동일한 검사, 처방이어도 환자 별로 수행 날짜가 다르고, 환자 별로 여러 번의 검사를 받는 경우에는 각 정보가 날짜 별로 하나의 레코드로 구분되어 테이블에 저장된다. 이로 인해, 어떤 환자의 정보를 검색할 때, 많게는 수십 개의 레코드를 확인해야 하고, 환자의 정보간의 상관 관계를 파악하기 어렵다. 이를 해결하기 위해 날짜별로 구분된 여러 레코드들을 하나의 레코드로 합치고 시계열 형태로 변환하여 데이터 관리 및 처리의 효율을 높였다. 본 절에서는 "검사", "처방", "수술" 테이블을 시계열 데이터로 변환하는 과정을 소개한다.

"검사" 테이블에는 환자의 각종 검사 내역들이 기록되어 있다. 급성 신장 손상 발병 판단에는 최소 2일~최대 7일을 필요로 하므로, 환자 별로 입원 후 7일간의 검사 내역들만 추출하였다. 또한, 입원 전 6개월 내에 수행했던 검사 또한 결과에 영향을 줄 수 있을 거라는 전문가의 조언에 따라 입원 전 6개월 내에 검사 내역 중 가장 최근 내역을 추출하였다. 즉, 환자 별로 총 8일 간의 검사 내역 정보를 가진다. 변환된 시계열 테이블은 그림 2와 같이 추출한 시계열 정보에 대해 입원 전 6개월 내의 검사 내역은 접두사 "pre"를 붙여 구분하고, 입원 후 1~7일 내역은 접두사 "d[n]"을 붙여서 구분하였다.

"검사" 데이터의 시계열 변환 알고리즘은 그림 3과 같다. 먼저, 환자의 입원 날짜 정보 참고를 위해 "환자 정보" 테이블과 "검사" 테이블을 조인하여 필요한 내역들을 추출한다(2-6). 각 레코드별로 입원일과 검사일의 차이를 계산하고, 그 차이에 따라서 입원 이전 혹은 입원 후 n일차 검사 내역 컬럼에 동일한 검사 내역 결과를 입력한다(7-21). 만약 검사 내역이 없다면 null 값이 기록되어 있을 것이다. "처방" 데이터의 시계열 변환 과정도 이와 동일하다.

수술 데이터의 시계열 변환 과정은 검사 데이터와 동일하지만, 전문가의 조언에 따라 수술 시간과 마취 정보만을 활용하기로 결정하였다. 이 때, 수술 시간과 마취 종류에 따라서 이산화하는 작업도 같이 수행하였다. 변환 알고리즘은 그림 4에 나타내었다.

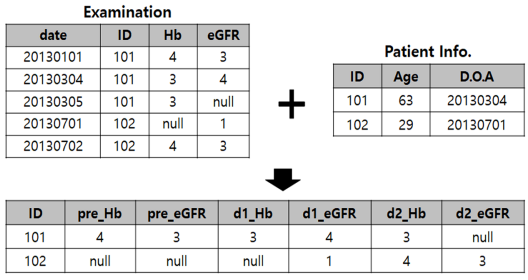


그림 2 시계열 변환의 예
Fig. 2 Time series transformation example

Algorithm 2 Transform to Series

```
Method OperationToSeries()
  RES res ← SQL('SELECT *
                 FROM
                   SELECT ID, DOA FROM Info
                   JOIN Ex
                 USING(ID)
  FOR EACH row IN res:
  INT diff ← DOA - Ex_Date
    IF diff ≤ 180:
      FOR EACH column IN row:
        SQL('UPDATE Ex_Series
             SET pre + 'colname(column)' + '= column')
    ELSE IF diff == -1:
      FOR EACH column IN row:
        SQL('UPDATE Ex_Series
             SET d1_+ 'colname(column)' + '= column')
    ...
    ELSE IF diff == -7:
      FOR EACH column IN row:
        SQL('UPDATE Ex_Series
             SET d7_+ 'colname(column)' + '= column')
```

그림 3 시계열 변환

Fig. 3 Time series transformation

Algorithm 3 Transform to Series(Operation)

```
Method ToSeries()
  RES res ← SQL('SELECT *
                 FROM
                   SELECT ID, DOA FROM Info
                   JOIN OP
                 USING(ID)
  INT diff ← DOA - OP_date
    IF diff ≤ 180:
      IF row.OP_time ≥ 1:
        SQL('UPDATE OP_Series SET pre_major = 1')
      ELSE:
        SQL('UPDATE OP_Series SET pre_minor = 1')
      IF row.Anes. == 'general':
        SQL('UPDATE OP_Series SET pre_general = 1')
      ELSE:
        SQL('UPDATE OP_Series SET pre_non-general = 1')
    ...
    ELSE IF diff ≤ -7:
      IF row.OP_time ≥ 1:
        SQL('UPDATE OP_Series SET d7_major = 1')
      ELSE:
        SQL('UPDATE OP_Series SET d7_minor = 1')
      IF row.Anes. == 'general':
        SQL('UPDATE OP_Series SET d7_general = 1')
      ELSE:
        SQL('UPDATE OP_Series SET d7_non-general = 1')
```

그림 4 수술 데이터 변환

Fig. 4 Operation data transformation

제안한 알고리즘은 “검사” 데이터 변환 과정과 유사하다. 먼저, 환자의 입원 날짜 정보 참고를 위해 “환자 정보” 테이블과 “수술” 테이블을 조인하여 필요한 내역들을 추출하고(2-6), 입원일과 수술일의 차이에 따라서 수술 정보를 시계열 형태로 저장한다. 수술 시간이 1 시간을 초과하는 경우 “major”, 아닌 경우 “minor”로 이산화하고, 마취 종류에 따라 전신 마취인 경우 “general”, 아닌 경우 “non-general”로 구분하였다.

3.5.2 크레아티닌 baseline 추가

급성 신장 손상 발병을 판단하기 위해서는 크레아티닌 수치의 변화량을 파악해야 하는데, 이를 위해선 환자의 크레아티닌 baseline 정보가 필요하다. 크레아티닌 baseline으로는 1) 입원 전 14일 이내 2) 90일 이내 3) 180일 이내 4) 입원 후 48시간 이내 검사 결과를 후보로 하며, 각 구간 별로 최소 값만 사용한다. 선정 기준은 값의 크기는 상관없이 명시된 순서를 기준으로 우선적으로 선출한다. 예를 들면, 입원 전 14일 이내 검사 내역이 있다면 90일 이내의 검사 내역 중 더 작은 값이 있어도 확인하지 않고 14일 이내의 검사 내역 중 최소 값을 baseline으로 사용한다. 검사 내역이 존재하지 않는 환자의 데이터는 분석에서 제외하였다.

표 4 진단 코드 통합 기준

Table 4 Diagnosis code integration criteria

Code	Disease	Result
I20 – I25	Ischemic heart disease	IH
I10 – I15	Hypertension	HT
E10 – E14	Diabetes	diabetes
I50	Heart Failure	HF
K*	Liver cirrhosis	liver
I60 – I69	Cerebrovascular disease	cereb
J44	COPD	COPD
N18	CKD	CKD
N17	AKI	AKI
C00 – C97	Cancer	cancer
B20 – B24	HIV	HIV

3.5.3 컬럼 이진 변환

진단 데이터에는 의료 데이터 특성 상 질병을 상세하게 구분하였는데, 이런 경우 해당 정보가 너무 지엽적이게 되어 분석 결과에 큰 영향을 주지 못한다. 이를 해결하기 위해, 표 4의 기준에 따라, 기존 값들을 보다 일반적인 11가지 질병으로 다시 분류하였다. 또한, 환자 별로 여러 개의 질병을 진단받는 경우가 있었기 때문에, 각 질병 별로 이진 변수 0,1 값을 가지는 컬럼을 생성하여, 특정 질병을 가지는 경우 해당 컬럼에 1, 아닌 경우 0을 입력하는 식으로 데이터를 변환하였다. 이 때, 입원 후에 진단받은 내역들의 정보는 모두 제거하였다.

“수술” 테이블에 대해서도 수술 시간과 마취 정보에 대한 이진 컬럼을 생성하여, 각 경우에 해당하면 해당 컬럼에 1, 해당하지 않으면 0을 입력하였다.

3.6 최종 데이터

3.6.1 최종 데이터 통합

본 절에서는 앞에서 처리된 테이블들을 최종 테이블로 통합한 결과를 보인다. 최종 테이블은 “환자 정보(Info.)” 테이블, 이진화된 “진단(Dx)” 테이블 그리고 시계열 변환된 “검사(Ex)”, “처방(Rx)”, “수술(OP)” 테이블들을 환자 ID를 기준으로 조인한 결과이다. 최종 데이터의 형태는 그림 5와 같다.

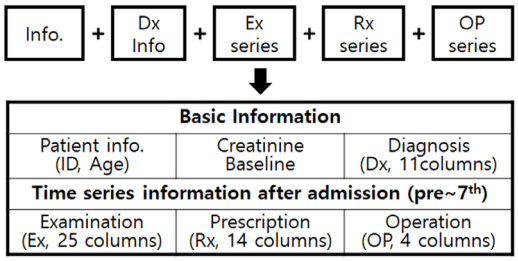


그림 5 통합 데이터 스키마

Fig. 5 Integrated data schema

3.6.2 최종 통합 데이터 클리닝

환자들이 매일 모든 검사, 처방 및 수술을 받는 것이 아니기 때문에, 입원 경과일 별로 구분된 최종 통합 데이터는 다수의 결측치(Missing value)를 포함한다. 예를 들어, 그림 6에서 환자 101은 입원 2일차에만 크레아티닌 검사를 수행하였기 때문에 다른 일자에는 크레아티닌 검사 결과가 누락되어 있다. 이러한 결측치는 전문가의 조언에 따라 누락된 정보를 전날 정보로 대체하고, 만약에 입원 1일차 정보가 누락된 경우에는 입원 전 6개월 내의 정보로 대체하는 방식으로 처리하였다.

하지만, 위 결측치 대체 방법은 입원 전후 정보가 모두 누락된 경우엔 대체할 값이 없기 때문에 여전히 결측치가 남게 된다. 본 연구에서 사용한 데이터는 약 17만건으로 충분히 많기 때문에, 결측치 대체 후에도 결측치가 남은 레코드는 모두 제거하였다.

ID	pre_Cr	d1_Cr	d2_Cr	d3_Cr
101	6.5	null	7.5	null

ID	pre_Cr	d1_Cr	d2_Cr	d3_Cr
101	6.5	6.5	7.5	7.5

그림 6 시계열 데이터 결측치 대체의 예

Fig. 6 Missing value imputation of time series data

3.6.3 최종 데이터 축소

전문가의 의견에 따라, 최종 데이터에서 예측 모델에 도움이 되지 않는 레코드들을 일부 제거하였다. 나이가 만 18세 이하이거나 투석 중인 환자, 크레아티닌 baseline이 4이상이거나 정보가 잘못 기입된 환자들의 레코드를 모두 제거하였다.

3.6.4 target 변수 생성

급성 신장 손상 발병 여부는 분석의 target 변수이지만, 그 값을 크레아티닌의 변화량으로 판단하기 때문에 주어진 데이터에는 발병 여부를 나타내는 변수가 존재하지 않는다. 따라서, 크레아티닌 baseline 값과 크레아티닌

Algorithm 5 Check Acute Kidney Injury

```
1:  Method CheckAKI()
2:    RES res ← SQL('SELECT ID, base
3:                   FROM CrBaseline
4:      FOR EACH row IN res:
5:        RES CrSeries ← SQL('SELECT d1_Cr, d2_Cr, ... , d7_Cr
6:                            FROM Integrated
7:        INT AKIres = 0
8:        IF d1_Cr > res.base + 0.3 OR d1_Cr > res.base*1.5:
9:          AKIres = 1
10:       ELSE IF d2_Cr > res.base + 0.3 OR d2_Cr > res.base*1.5:
11:         AKIres = 2
12:         ...
13:       ELSE:
14:         AKIres = 7
15:       IF AKI > 0:
16:         SQL('UPDATE Integrated
17:              SET AKI = AKIres
18:              WHERE id = '+res.ID')
```

그림 7 급성 신장 손상 발생일 변수 생성

Fig. 7 AKI occurrence date creation

검사 결과를 활용하여 환자 별로 급성 신장 손상 발생 여부를 나타내는 새로운 변수를 계산해주어야 한다. 급성 신장 손상 발생일을 계산하는 알고리즘은 그림 7과 같다.

제안하는 급성 신장 손상 발생 판명 알고리즘은 통합 테이블에서 급성 신장 손상이 입원하고 며칠 뒤에 발생했는지 계산하여 기록한다. 먼저, 3.5.2에서 계산한 크레아티닌 baseline 값을 불러오고(2-3), 크레아티닌 baseline 값이 존재하는 사람에 한해서 입원 1~7일차에 해당하는 크레아티닌 검사 수치를 불러온다(4-6). 날짜 별로 baseline과 비교하고, 급성 신장 손상이 입원 후 1일차에 발생하면 1, 3일차에 발생하면 3을 입력하는 식으로 처리한다(9-14). 급성 신장 손상이 발생한 사람들의 정보를 갱신해주고 알고리즘을 종료한다.(15-18).

3.6.5 최종 데이터 변환

기존 데이터는 의료 데이터를 시계열 형태로 저장하여 입원 일을 기준으로 며칠 뒤에 급성 신장 손상이 발생했는지에 대한 정보를 담고 있다. 하지만, 본 연구의 목적은 하루치 데이터로 다음 날 급성 신장 손상 발생 여부를 예측하는 것이므로, 기존 데이터를 그림 8과 같이 문제에 적합한 형태로 변환한다. 먼저, 3.6.4에서 생성한 급성 신장 손상 발생일 변수를 1~8일차까지 8개 이진 컬럼으로 변환하고, 급성 신장 손상 발생시 각 컬럼에 1을 입력하였다. 생성된 이진 컬럼들은 n일차 입원 데이터에 추가하여 환자 별로 8일 동안의 정보를 1일 단위로 분리하였다. 8일간의 데이터 중 급성 신장 손상이 발생한 날 이후의 데이터는 이미 질병이 발생하고 나서 기록된 정보이기 때문에 예측에 도움이 되지 않으므로 제외하였다. 또한, 입원 후 일주일이 지나기 전에 퇴원하는 환자들이 많았는데, 이 경우 퇴원일 이후의 정보들도 제외하였다. 그 결과, 최종 데이터는 환자 78,935명에 대한 190,985건의 레코드를 포함한다.

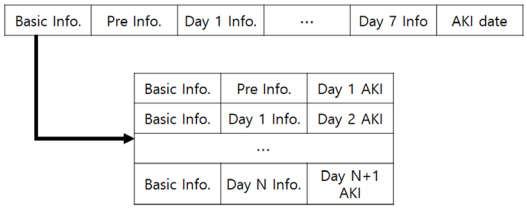


그림 8 시계열 데이터 분리의 예
Fig. 8 Splitting time series data example

4. 실험

4.1 오버 샘플링

본 연구에서는 3장에서 소개한 전처리 작업의 적합성을 확인하기 위하여, 3개의 기계 학습 알고리즘을 활용하여 간단한 실험을 진행하였다. 실험에는 전처리된 분당 서울대병원 환자 데이터를 사용하였는데, 전처리 결과 최종 데이터는 환자 78,935명에 대한 190,985건의 레코드를 포함하며 58개의 컬럼으로 구성된다. 데이터의 클래스 분포는 급성 신장 손상 발생이 21,486건, 미발생이 169,499건으로 클래스간 비율이 대략 1:9인 불균형 데이터이다. 이러한 불균형 데이터로 예측 모델을 학습하면 모델이 다수 클래스에 과적합되어 정확도(Accuracy)는 높지만 재현율(Recall)은 감소하여, 모델의 실효성이 크게 떨어지게 된다. 이런 문제는 모델이나 데이터의 수정을 통해 해결 가능한데, 그 예시로는 모델의 threshold 조정, 오버샘플링 혹은 언더샘플링을 통한 인위적인 데이터 분포 조절 방법 등이 있다.

본 연구에서는 실험을 진행하기 전 오버샘플링 방법을 사용하여 클래스 불균형 문제를 해결하였다. 알고리즘은 오버샘플링 알고리즘 중에서도 양호한 성능을 낸다고 알려져 있는 SMOTE[13]를 사용하였다. SMOTE는 소수 클래스 데이터를 생성하여 데이터의 수를 증가시키는 대표적인 오버샘플링 알고리즘으로, K-Nearest Neighbors의 개념을 활용하여 새로운 데이터를 생성할 때 소수 클래스들의 특징을 반영하도록 한다.

본 연구에서는 SMOTE로 target 변수의 클래스 비율을 1:1로 맞춘 후에 예측 모델 학습에 사용하였다. 오버샘플링은 학습 데이터에만 적용하여 과적합된 모델을 생성하는 것을 방지하였고, 오버샘플링을 적용하지 않은 테스트 데이터로 검증하여 모델의 실효성을 검증하였다. 실험에 사용한 SMOTE 알고리즘은 Python의 imbalanced learn 모듈에서 제공하는 알고리즘을 사용하였다.

4.2 실험

실험은 Windows 10 단일 노드에서 진행하였으며, 노드의 사양은 Intel® Core i7-7700 3.10Ghz, 16GB RAM이다. 실험에 사용한 3가지 분석 모델은 1) Logistic Regression 2) Random Forest 3) MLP(Multi-layer Perceptron)이며 모든 분석 모델은 Python의 scikit-learn 모듈에서 제공하는 알고리즘을 사용하였다. 분석 모델들의 실험 결과는 기존에 서울대병원에서 급성 신장 손상 예측에 사용하던 SPARK의 결과와 함께 비교하였다. 실험에 사용한 데이터 중 임의로 선택한 2/3는 학습용으로 지정하여 오버샘플링을 적용하였고, 나머지 1/3은 테스트용으로 사용하였다. 분석 결과의 주요 평가 지표로는 ROC(Receiver Operating Characteristics)의 AUC (Area Under the Curve)을 사용하였는데, ROC AUC는 전체적인 민감도(Sensitivity)와 특이도(Specificity)의 상관 관계를 보여줄 수 있어, 클래스가 불균형하여 정확도가 큰 의미가 없는 경우에 사용하기 적합한 척도이기 때문이다.

표 5 예측 결과 비교
Table 5 Prediction results comparison

	LR	RF	MLP	SPARK
Accuracy	0.75	0.92	0.77	0.631
Sensitivity	0.689	0.338	0.748	0.545
Specificity	0.751	0.926	0.77	0.632
AUC	0.805	0.766	0.834	0.624

각 분석 모델의 중요 파라미터 설정은 다음과 같다. Random Forest에서는 총 100개의 Tree를 생성하는데, 각 Tree는 Gini 계수를 기준으로 분기하며, 최대 깊이는 10으로 제한하였다. MLP의 경우 히든 레이어는 2층으로, 첫 번째 레이어는 5개, 두 번째 레이어는 4개의 노드로 구성하였고, 활성 함수는 ReLU, 최적화 알고리즘은 Adam을 사용하였다. 이 외에는 별도의 최적화 작업이나 파라미터 조정을 하지 않고 실험을 진행하였다.

분석 모델 별 예측 실험 결과는 표 5에 정리하였다. 먼저, 분석 모델들 중에서는 MLP가 가장 좋은 성능을 보임을 확인하였다. Random Forest 모델이 정확도와 특이도에서 굉장히 좋은 결과를 기록하였지만, 반대로 민감도가 굉장히 낮은 걸로 보아 모델이 특정 클래스에 과적합되었음을 알 수 있으며, AUC에서도 상대적으로 낮은 수치를 기록하였다.

SPARK와 비교했을 때 3가지 분석 모델 모두 더 좋은 성능을 보였는데, 이를 통해 본 연구에서 제안한 전처리 작업의 결과가 일반적인 기계 학습 모델에 활용되기에 적합했다고 판단할 수 있는 수치이다. 또한, 본 실험의 결과가 관련 연구들 중에서 가장 높은 수치를 기록한 [4]보다는 성능이 좋지 않지만, [4]이 소급 적용하는 방식으로 실험하였다는 점을 고려하면 충분히 유의미한 결과이다.

5. 결 론

본 논문에서는 정돈되지 않고 파편화된 의료 데이터

를 분석 모델에 활용할 수 있는 형태로 정제하는 체계적이고 효율적인 전처리 알고리즘을 제안하였다. 전처리 작업을 수행하기에 앞서, 다양한 분과 및 담당의사 별로 구분되어 수집된 수백 개의 파일을 분석하여 데이터에 내재된 특성을 파악하였다. 이어서, 다섯 가지 전처리 작업인 데이터 클리닝, 통합, 변환, 축소 그리고 이산화를 통해 파악한 특성이 잘 드러나도록 데이터를 정제하였다. 마지막으로, 간단한 실험을 통해 제안한 전처리 작업을 적용한 데이터가 분석 모델 학습에 유효하고 효과적으로 작용함을 확인하였다.

본 연구는 데이터 전처리 작업에 초점을 맞추었기 때문에 그 외의 요소들에 대해서는 크게 고려하지 않았다. 따라서, 중요 변수 선택, 분석 모델 선정 혹은 모델 파라미터 수정 등의 작업을 통해 예측 성능을 향상시킬 여지가 남아있다. 또한, 기존에 수집되지 않은 정보 중에서도 중요한 정보가 있을 수 있기 때문에 정보를 추가 수집한 후 모델의 성능 향상을 위한 연구를 진행할 예정이다. 마지막으로, 데이터에 적합한 모델 및 파라미터를 파악하는 방식으로 의료 데이터 마이닝을 위한 자동화된 데이터 수집 및 축적 시스템 설계 및 개발을 수행할 것이다.

References

[1] D. Kim, K. Joo, "Definition and Evaluation of Acute Kidney Injury: Clinical Practice Guidelines," *The Korean Journal of Medicine*, Vol. 88, No. 4, 2015.

[2] Kate, R. J., Perez, R. M., Mazumdar, D., Pasupathy, K. S., & Nilakantan, V. Prediction and detection models for acute kidney injury in hospitalized older adults. *BMC medical informatics and decision making*, 16(1), 39, 2016.

[3] Lee, Hyung-Chul, et al., "Derivation and Validation of Machine Learning Approaches to Predict Acute Kidney Injury after Cardiac Surgery," *Journal of clinical medicine*, 7.10, 2018.

[4] Mohamadlou, Hamid, et al., "Prediction of Acute Kidney Injury With a Machine Learning Algorithm Using Electronic Health Record Data," *Canadian Journal of Kidney Health and Disease 5*, 2018.

[5] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *Applied Artificial Intelligence*, Vol. 17, No. 5-6, pp. 375-381, 2003.

[6] D. Tanasa and B. Trousse, "Advanced data preprocessing for intersites web usage mining," *Intelligent Systems*, IEEE, Vol. 19, No. 2, pp. 59-65, 2004.

[7] H. Jun, G. Hyun, K. Lim, W. Lee, and H. Kim, "Big Data Preprocessing for Predicting Box Office Success," *KIISE Transactions on Computing Practices*, Vol. 20, No. 12, pp. 615-622, 2014.

[8] Huang, M. W., Lin, W. C., Chen, C. W., Ke, S. W., Tsai, C. F., & Eberle, W., "Data preprocessing issues for incomplete medical datasets," *Expert Systems*, Vol. 33, No. 5, pp. 432-438, 2016.

[9] Almuhaideb, S., Menai, M. E. B., "Impact of preprocessing on medical data classification," *Frontiers of Computer Science*, Vol. 10, No. 6, pp. 1082-1102, 2016.

[10] A. Werner, M. Bach, and W. Pluskiewicz, "The study of preprocessing methods' utility in analysis of multidimensional and highly imbalanced medical data," *Proceedings of 11th International Conference IIIS2016*, 2016.

[11] A. P. Hassler, et al., "Importance of medical data preprocessing in predictive modeling and risk factor discovery for the frailty syndrome," *BMC medical informatics and decision making*, Vol. 19, No. 1, 33, 2019.

[12] W. Zhang, [Online]. Available: http://www.cs.wustl.edu/~zhang/teaching/cs514/Spring11/Data-prep.pdf (Accessed: 10 Feb 2019)

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of artificial intelligence research*, Vol. 16, No. 1, pp. 321-357, Jan. 2002.

양 현 식

2018년 전북대학교 IT정보공학과 학사 2018년~현재 서울대학교 컴퓨터공학부 석사과정 재학 중. 관심분야는 데이터 마이닝

임 유 빈

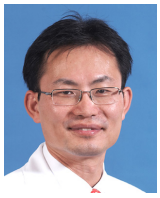
2015년 한동대학교 전산전자공학부 학사 2015년~현재 서울대학교 컴퓨터공학부 박사과정 재학 중. 관심분야는 데이터베이스, 데이터 마이닝

이 진 영

2006년 서울시립대학교 국어국문학과 학사. 2019년~현재 퓨전데이터 데이터 분석가. 관심분야는 공공 및 금융 데이터 마이닝

김 동 효

2016년 홍익대학교 컴퓨터공학부 학사 2016년~2019년 8월 서울대학교 컴퓨터공학부 박사과정 수료. 관심분야는 데이터베이스, 데이터 마이닝



김 세 중

1998년 서울대학교 의과대학 의학과 학사 2003년 서울대학교 내과학 석사. 2008년 서울대학교 내과학 박사. 2019년~현재 분당서울대학교병원 신장내과 교수. 2018년~현재 대한신장학회 일반이사, KRCP 편집위원

김 형 주

정보과학회 컴퓨팅의 실제 논문지
제 25 권 제 1 호 참조