

갤럭시 스마트폰의 사용자

데이터 시각화

요약

스마트폰은 현재 삶의 필수품이라고 볼 수 있을 정도로 많은 사람들의 소유물이 되었다. 다만 이러한 스마트폰은 개인정보를 가장 많이 담고 있는 도구이며 이 때문에 범죄의 표적이 되기도 한다. 본 연구는 디지털 포렌식을 위한 갤럭시 스마트폰에 대한 데이터분석 및 수집 방법을 제안한다. 제안하는 방법은 갤럭시 스마트폰 포렌식의 기초 데이터를 만들고, 결론적으로 추출한 데이터를 시각화 하여 접근하기 쉽도록 제안한다.

1 서론

1.1 연구 배경

스마트폰이 등장하고 나서 스마트폰은 개인 정보를 가장 많이 담고 있는 도구가 되었고 이때문에 범죄의 표적이 되기도 한다. 또한 범죄에도 직접 쓰이기도 하는데 이런 범죄 활동을 법정에서 증명하는데 스마트폰은 결정적인 역할을 하기도 한다. 그 역할을 해줄 수 있게 해주는 기술을 디지털 포렌식이라고 한다. 디지털 포렌식은 디지털 증거물을 분석하여 수사에 활용하고, 디지털 증거물의 증거 능력을 향상시키기 위한 과학 수사 기법을 총칭하는 용어이다. 포렌식 기술은 앞으로 점점 스마트폰을 이용한 범죄에 필수적인 기술로 자리 잡을 예정이며, 더 많은 연구가 요구되고 있다.

포렌식 도구의 개발에 대한 요구는 꾸준히 늘고 있으며, 데이터를 복구하기 위해서는 연구가 필요하다고 판단이 되었다. 따라서 안드로이드 운영체제에서 데이터를 수집, 분석을 하는 포렌식 도구를 개발하려고 한다.

본 연구에서는 데이터를 수집, 분석을 하는 기존의 포렌식 방식을 안드로이드 폰인 '갤럭시' 시리즈에 적용해보는 연구를 진행할 예정이다.

1.2 연구 목표

안드로이드 운영체제를 사용하는 '갤럭시' 시리즈 폰의 데이터를 수집, 분석하고 분석한 데이터를 통해 어떠한 데이터가 시스템 데이터에 들어있는지를 시각화 하는 작업을 진행할 예정이다.

이를 위해 안드로이드 시스템 배경, ADB 아키텍처 및 디지털 포렌식의 기본 원칙을 포함하여 시스템 연구를 진행할 것이며, 시스템 데이터를 얻기 위해 갤럭시 폰에 루팅을 진행한 후 데이터를 추출, 분석할 계획이다. 이 데이터를 시각화 하는 작업은 react를 통해 웹으로 보여줄 예정이며, 모든 사용자 데이터를 보여주는 것이 아닌 특정 사용자 데이터를 선정 후 웹으로 보여줄 예정이다.

2 관련연구

2.1 덤프

갤럭시 핸드폰의 파일시스템을 통째로 파일로 추출하는 방법이다. 파일시스템 뿐만 아니라 다른 여러 정보 또한 같이 추출한다.

2.2 루팅

루팅은 모바일 기기에서 구동되는 안드로이드 운영 체제 상에서 최상위 권한을 얻음으로 해당 기기의 생산자 또는 판매자 측에서 걸어 놓은 제약을 해제하는 행위를 가리키는 말이다. 이 루팅을 통해서 데이터에 접근이 가능해지기 때문에 갤럭시 폰에 대한 루팅이 필요하다.

2.3 ADB

안드로이드 장치와 통신하여 디버깅 등의 작업을 진행할 수 있는 Command line tool 안드로이드 SDK에도 포함되어 있으며 애플리케이션 설치, 디바이스 접속 및 관리, 파일 업/다운로드, 시스템 log 출력, shell 접속 등이 가능하다. 따라서 접근이 제한되어 있는 애플리케이션의 데이터에 대해 복원, 추출이 가능하게 해준다.

2.4 Smart Switch

스마트 스위치는 기존 데이터에 영향을 주지 않고, 원래 스토리지에 백업 데이터를 추가로 저장하는 형식으로 작동한다. PC와 모바일 환경 모두 사용할 수 있으며 설치형 프로그램이 아니어서 다운로드 한 후 바로 실행할 수 있다. 페어링만 제대로 하면 별다른 설정 없이 기기간 데이터 이동이 이뤄지는 것도 특징이다. 단말간 연결을 위한 정보 SSID나 PW를 하이프리퀀시 오디오로 전송하여 S빔과 유사하게 스마트폰을 맞대면 바로 페어링 할 수 있다. 동시에 NFC와 Wi-Fi Direct도 지원하고 있다. 이외에도 삼성 계정, 구글 계정, 각종 클라우드와 연계해서 실행되기도 한다.

2.5 React

리엑트는 자바스크립트 라이브러리의 하나로서 사용자 인터페이스를 만들기 위해 사용된다. 페이스북과 개별 개발자 및 기업들 공동체에 의해 유지보수된다. 리엑트는 싱글 페이지 애플리케이션

션이나 모바일 애플리케이션 개발에 사용될 수 있다

2.6 TypeScript

타입스크립트(TypeScript)는 자바스크립트의 슈퍼셋인 오픈소스 프로그래밍 언어이다. 마이크로소프트에서 개발, 유지하고 있으며 엄격한 문법을 지원한다. C#의 리드 아키텍트이자 델파이, 터보 파스칼의 창시자인 Anders Hejlsberg가 개발에 참여한다. 클라이언트 사이드와 서버 사이드를 위한 개발에 사용할 수 있다. 타입스크립트는 자바스크립트 엔진을 사용하면서 커다란 애플리케이션을 개발할 수 있게 설계된 언어이다. 자바스크립트의 슈퍼셋이기 때문에 자바스크립트로 작성된 프로그램이 타입스크립트 프로그램으로도 동작한다. 타입스크립트에서 자신이 원하는 타입을 정의하고 프로그래밍을 하면 자바스크립트로 컴파일되어 실행할 수 있다. 타입스크립트는 모든 운영 체제, 모든 브라우저, 모든 호스트에서 사용 가능한 오픈 소스이다.

2.7 APEXCHARTS

ApexCharts는 웹 페이지에 차트를 보여주는 차트 라이브러리이다. 선 그래프, 도넛형, 히트맵 등 다양한 스타일들이 있고, 반응형에 커스텀 기능까지 구현되어 있다. MIT 라이선스를 받은 오픈 소스라서 상용 어플리케이션에서도 사용이 가능하다.

2.8 기존 연구의 문제점 및 해결 방안

2.8.1 연구의 문제점

루팅한 데이터를 추출 후 분석을 했는데 루팅을 하고 나니 사용자의 모든 데이터가 초기화되어서 제한된 데이터를 분석을 해야 하는 환경이었다. 그래서 사용자 데이터 분석을 원활하게 하지 못했으며 어떤 데이터가 루팅 후에 추출되었는지 알기 쉽지 않았다.

2.8.2 해결 방안

루팅을 한 후 사용자 데이터에 접근이 가능해졌고, 이를 통해 어떤 사용자 데이터가 갤럭시 속에 있었는데 분석이 가능해졌다. 사용자 데이터를 분석 후 어떤 데이터가 갤럭시에 있는지 루팅 전후로 나누어서 루팅을 하면 얻을 수 있는 데이터에 대해 시각화를 진행할 예정이다. 또한 데이터 초기화 없이 데이터를 추출할 수 있는 프로그램인 Smart Switch에서 추출한 데이터를 파싱하여 시각화를 진행할 예정이다.

3. 프로젝트 내용

3.1. 시나리오

3.1.1. 파일시스템 덤프

루팅을 하지 않은 핸드폰에 대해 접근할 수 있는 파일 시스템에 한계점이 있다. 가장 개인정보가 많이 저장되어 있는 시스템에 접근하기 위해선 권한이 부족한 상태이다.

항 목	파일 경로
커널 정보	/data/log/recovery_kernel_log.txt
리커버리 로그	/data/log/recovery_log.txt
전원 종료 로그	/data/log/poweroff_info.txt
전원 재시작 로그	/data/log/powerreset_info.txt
전원 부팅 로그	/data/log/rtc.log
통화중 단절 로그	/data/log/CallDropInfoLog.txt
앱 에러 로그	/data/log/dumpstate_app_error.txt.gz
공유기 연결 로그	/data/misc/wifi/wpa_supplicant.conf
블루투스 정보	/data/misc/bluetoothd/config
앱 실행 정보	/data/system/dmappmgr.db
설치된 앱 정보	/data/system/packages.xml
등록된 계정 정보	/data/system/users/0/accounts.db
자동 로그인 계정 정보	/data/system/registered_services/ android.accounts.AccountAuthenticator.xml
자동 동기화 앱 목록	/data/system/registered_services/ android.content.SyncAdapter.xml

그림 1 안드로이드 모바일기기의 시스템 데이터 제목 및 파일 경로

[그림 1]은 기존 연구 중 안드로이드 모바일 기기의 시스템 데이터 파일 경로를 나타내고 있다 [2]. 안드로이드 버전이 올라감에 따라 내부 데이터 접근이 더욱 어려워졌으며, ADB 툴을 이용하여 데이터를 추출하는 데 한계가 있다.

본 장에서는 2가지 문제를 제시한다. 첫째는 루팅 없이 갤럭시 스마트폰에 접근할 수 있는 데이터는 한정되어 있다는 점이다. 두 번째 문제는 ADB를 통해 수집할 수 있는 데이터를 수집해본 결과, 실제 데이터에 접근할 수 없도록 암호화가 되어 있는 파일들이 존재한다는 점이다.

첫 번째 문제를 해결하기 위해 여러 가지 해결 방안을 고민해볼 수 있다. 루팅을 통해 루트 권한을 획득하는 방법, 물리적으로 메모리를 추출하여 파일시스템에 접근하는 방법, Smart Switch(삼성 스마트폰 백업 프로그램)의 데이터를 하이재킹하여 데이터를 수집하는 방법 등을 제시할 수 있다.

루팅을 통해 루트 권한을 획득할 경우 가장 쉽게 루트권한을 취득할 수 있다는 장점이 있다. 다만 안드로이드 버전 업에 따라 루팅을 할 수 있는 커널이 업데이트 될 때까지 기다려야 한다는 단점이 있으며, 또한 추후 포렌식 진행 시 데이터의 무결성에 문제가 생길 수도 있다.

두 번째로 물리적으로 메모리를 추출하여 파일시스템에 접근하는 방법은 추가 작업이 필요 없이 메모리를 추출하면 되기 때문에 시간적으로 가장 빠르다는 장점이 있다. 다만 PUF(Physical Unclonable Function)이 적용되어 있는 갤럭시 스마트폰의 경우 물리적인 추출을 진행할 경우 데이터가 모두 손상될 수 있다는 단점이 있다.

마지막으로 Smart Switch(삼성 스마트폰 백업 프로그램)의 데이터를 하이재킹을 할 경우, 물리적, 논리적 손상 없이 데이터를 추출할 수 있다는 장점이 있다. 다만 시스템 데이터에는 접근이 불가하여 데이터 추출에 제한이 있다는 단점이 있다.

아래 [표 1]은 스마트폰 데이터 접근 방안별 장단점을 정리한 표이다.

표 1. 스마트폰 데이터 접근 방안별 장단점

방안	장점	단점
루팅	쉽게 루트권한 취득 가능	데이터가 초기화 되어서 분석에 한계가 생김
물리적 추출	시간적으로 가장 빠르며, 쉽게 파일 시스템 접근 가능	PUF의 존재 유무에 따라 데이터 손상 가능
Smart Switch	물리적, 논리적 손상 없이 데이터 추출 가능	암호화되어 있는 파일이 존재

두 번째 문제를 해결하기 위해 분석 대상 스마트폰을 추가 분석하여 실제로 얻어낼 수 있는 데이터들을 파악할 수 있다. 예를 들면 기본 인터넷 브라우저의 즐겨찾기 등의 사용자가 쉽게 접근할 수 있는 데이터를 기준으로 암호화되어 있는 데이터에 접근하여 추가 분석을 진행한 뒤 암호화되어 있는 데이터를 복호화 할 수 있다.

3.1.2. 파일시스템 추출

3.1.2.1 루팅

루팅한 데이터에서 나온 로그들과 데이터를 분석하여 어떤 정보가 로그와 데이터에 들어있는지 분석을 하고 이 데이터가 루팅을 해야만 얻을 수 있는 데이터인지 파악을 할 것이다. 아래의

분석 결과는 루팅 후 Dump 를 통해 얻은 로그들과 데이터베이스 파일을 분석하여 주요 파일을 표로 정리한 것이다.

주요 로그 파일 분석 결과

파일명	저장 데이터
power_off_reset_reason.txt	전원 관련 로그 저장
qtables.json	네트워크 접근 기록 저장
lwc_dump.txt	WIFI 접근 기록 저장
subBuffer.log	블루투스 접근 기록 저장
CallContent.log	전화 기록 저장
Recovery_history.log	복구 기록 저장
Settingsprovider.txt	핸드폰의 세팅 변경 기록 저장
Shutdown_profile.n.txt (n은 1이상 숫자)	기기의 종료 기록 저장
Packages.xml	어플리케이션의 목록 및 각각의 권한 저장
n/Settings_config.xml (n은 0이상 숫자)	사용자별 기기 세팅 정보 저장

주요 데이터베이스 파일 분석 결과

파일명	저장 데이터
Audioservice_sec.db	어플리케이션의 오디오 권한 저장
ClipboardimageTable.db	클립보드에 있는 스크린 샷 파일 목록 저장
Displaysolution_setting.db	화면 권한 설정 저장
Enterprise.db	기기의 전반적인 설정 저장
Gamemanager.db	게임 매니저에 관한 앱과 로그 저장
Locksettings.db	화면 잠금에 대한 정보 저장

Notification.db	알림들의 로그 저장
Pda.db	단말기 정보 저장
Pkgpredictions.db	패키지 정보 저장
Psitracker.db	Psi 측정정보 저장
Recoverablekeystore.db	클라우드 정보 저장
wifihistory.db	와이파이 사용 기록 저장

3.1.2.2 Smart Switch

삼성에서 제공하는 어플리케이션이자 프로그램인 Smart Switch 로 데이터를 추출한 결과이다. 추출한 데이터는 JSON 형태로 되어 있으며, 이 데이터를 파싱하여 주요 데이터에 대해 시각화를 진행할 예정이다. 또한 파싱한 데이터에서 사용자의 감정을 알아내기 위해 감정분석을 진행할 예정이다.

SAFETYSETTING	2022-10-27(목) 오전 ...	파일 폴더	
SAMSGNOTE	2022-10-27(목) 오전 ...	파일 폴더	
SAMSGVOICERECORD	2022-10-27(목) 오전 ...	파일 폴더	
SBROWSER	2022-10-27(목) 오전 ...	파일 폴더	
SCHEDULESETTING	2022-10-27(목) 오전 ...	파일 폴더	
SHEALTH2	2022-10-27(목) 오전 ...	파일 폴더	
SHORTCUT	2022-10-27(목) 오전 ...	파일 폴더	
SMARTMANAGER	2022-10-27(목) 오전 ...	파일 폴더	
TASKEDGEANAL	2022-10-27(목) 오전 ...	파일 폴더	
Video	2022-10-27(목) 오전 ...	파일 폴더	
WALLPAPER	2022-10-27(목) 오전 ...	파일 폴더	
WEATHERSERVICE	2022-10-27(목) 오전 ...	파일 폴더	
WIFICONFIG	2022-10-27(목) 오전 ...	파일 폴더	
WORLDLOCK	2022-10-27(목) 오전 ...	파일 폴더	
backupHistoryInfo.xml	2022-10-31(월) 오후 ...	XML 문서	159KB
BackupResult.txt	2022-10-27(목) 오전 ...	텍스트 문서	1KB
ReqItemsInfo.json	2022-10-27(목) 오전 ...	JSON File	9KB
SmartSwitchBackup.json	2022-10-27(목) 오전 ...	JSON File	1,857KB
SmartSwitchBackup_back.json	2022-10-27(목) 오전 ...	JSON File	1,244KB

3.1.3. 데이터 파싱

3.1.3.1 연도와 달 별로 분류

연도와 달별 데이터 파싱 코드

```

} else if (Number(sms._readable_date.slice(0, 4)) === '2020') {
    sms_2020.current[Number(sms._readable_date.slice(6, 8)) - 1] += 1;
    if (Number(sms._readable_date.slice(6, 8)) === 1) {
        sms_2020_Month.current.Jan.push(sms);
        smsAddresses_2020.current.Jan.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 2) {
        sms_2020_Month.current.Feb.push(sms);
        smsAddresses_2020.current.Feb.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 3) {
        sms_2020_Month.current.Mar.push(sms);
        smsAddresses_2020.current.Mar.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 4) {
        sms_2020_Month.current.Apr.push(sms);
        smsAddresses_2020.current.Apr.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 5) {
        sms_2020_Month.current.May.push(sms);
        smsAddresses_2020.current.May.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 6) {
        sms_2020_Month.current.June.push(sms);
        smsAddresses_2020.current.June.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 7) {
        sms_2020_Month.current.July.push(sms);
        smsAddresses_2020.current.July.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 8) {
        sms_2020_Month.current.Aug.push(sms);
        smsAddresses_2020.current.Aug.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 9) {
        sms_2020_Month.current.Sep.push(sms);
        smsAddresses_2020.current.Sep.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 10) {
        sms_2020_Month.current.Oct.push(sms);
        smsAddresses_2020.current.Oct.push(sms._address);
    } else if (Number(sms._readable_date.slice(6, 8)) === 11) {
        sms_2020_Month.current.Nov.push(sms);
        smsAddresses_2020.current.Nov.push(sms._address);
    } else {
        sms_2020_Month.current.Dec.push(sms);
    }
}

```

연도와 달별로 파싱된 데이터

[illegible]

3.1.3.2 연락처 별로 분류

연락처별 데이터 파싱 코드

```

smsAddressCount.current.smsAddressCount_2020.Jan =
    smsAddresses_2020.current.Jan.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.Feb =
    smsAddresses_2020.current.Feb.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.Mar =
    smsAddresses_2020.current.Mar.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.Apr =
    smsAddresses_2020.current.Apr.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.May =
    smsAddresses_2020.current.May.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.June =
    smsAddresses_2020.current.June.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.July =
    smsAddresses_2020.current.July.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.Aug =
    smsAddresses_2020.current.Aug.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.Sep =
    smsAddresses_2020.current.Sep.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.Oct =
    smsAddresses_2020.current.Oct.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.Nov =
    smsAddresses_2020.current.Nov.reduce(addresses, {});
smsAddressCount.current.smsAddressCount_2020.Dec =
    smsAddresses_2020.current.Dec.reduce(addresses, {});

```

연락처별 파싱된 데이터

[illegible]

3.1.3.3 보낸 문자 별로 분류

보낸 문자 데이터 파싱 코드

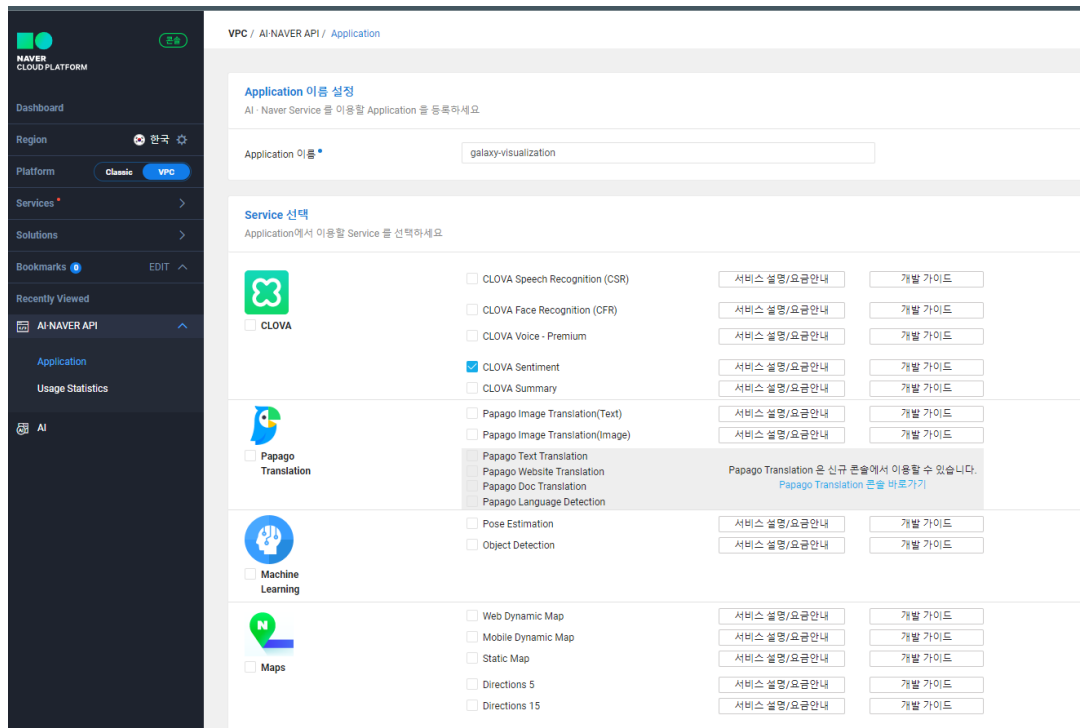
```
} else if (year === '2020') {  
  sms_2020_Month.current.Jan.forEach((sms) => {  
    if (sms._type === '2') bodyJan = bodyJan.concat(sms._body);  
  });  
  sms_2020_Month.current.Feb.forEach((sms) => {  
    if (sms._type === '2') bodyFeb = bodyFeb.concat(sms._body);  
  });  
  sms_2020_Month.current.Mar.forEach((sms) => {  
    if (sms._type === '2') bodyMar = bodyMar.concat(sms._body);  
  });  
  sms_2020_Month.current.Apr.forEach((sms) => {  
    if (sms._type === '2') bodyApr = bodyApr.concat(sms._body);  
  });  
  sms_2020_Month.current.May.forEach((sms) => {  
    if (sms._type === '2') bodyMay = bodyMay.concat(sms._body);  
  });  
  sms_2020_Month.current.June.forEach((sms) => {  
    if (sms._type === '2') bodyJune = bodyJune.concat(sms._body);  
  });  
  sms_2020_Month.current.July.forEach((sms) => {  
    if (sms._type === '2') bodyJuly = bodyJuly.concat(sms._body);  
  });  
  sms_2020_Month.current.Aug.forEach((sms) => {  
    if (sms._type === '2') bodyAug = bodyAug.concat(sms._body);  
  });  
  sms_2020_Month.current.Sep.forEach((sms) => {  
    if (sms._type === '2') bodySep = bodySep.concat(sms._body);  
  });  
  sms_2020_Month.current.Oct.forEach((sms) => {  
    if (sms._type === '2') bodyOct = bodyOct.concat(sms._body);  
  });  
  sms_2020_Month.current.Nov.forEach((sms) => {  
    if (sms._type === '2') bodyNov = bodyNov.concat(sms._body);  
  });  
  sms_2020_Month.current.Dec.forEach((sms) => {  
    if (sms._type === '2') bodyDec = bodyDec.concat(sms._body);  
  });  
}
```

3.1.4 감정 분석

3.1.4.1 CLOVA Sentiment API

추출한 데이터에서 사용자의 감정을 파악하기 위해서 네이버가 제공하는 CLOVA Sentiment API 를 사용하였다.

Naver Cloud Platform Console



CLOVA Sentiment API Body

필드명	필수 여부	타입	설명
document	필수	Object	전체 문장 관련 object
document.sentiment	필수	String	전체 문장에 대한 감정
document.confidence	필수	Object	전체 문장에 대한 감정 confidence
document.confidence.neutral	필수	Float	중립 confidence (%)
document.confidence.positive	필수	Float	긍정 confidence (%)
document.confidence.negative	필수	Float	부정 confidence (%)
sentences	필수	List of Object	분류 문장 관련 list object
sentences.content	필수	String	분류 문장
sentences.offset	필수	Int	document.content 에서 문장 시작 위치
sentences.length	필수	Int	분류 문장 글자 수
sentences.sentiment	필수	String	분류 문장 감정
sentences.confidence	필수	Object	분류 문장에 대한 감정 confidence
sentences.confidence.neutral	필수	Float	중립 confidence (%)
sentences.confidence.positive	필수	Float	긍정 confidence (%)
sentences.confidence.negative	필수	Float	부정 confidence (%)
sentences.highlights	필수	List of Object	sentences.content 에서 감정분석 구간
sentences.highlights.offset	필수	Int	주요 감정 구간 시작 위치
sentences.highlights.length	필수	Int	주요 감정 구간 글자 수
sentences.negativeSentiment.sentiment	필수	String	부정 감정일 경우, 세부 감정
sentences.negativeSentiment.confidence	필수	Float	부정 감정일 경우, 세부 감정에 대한 confidence

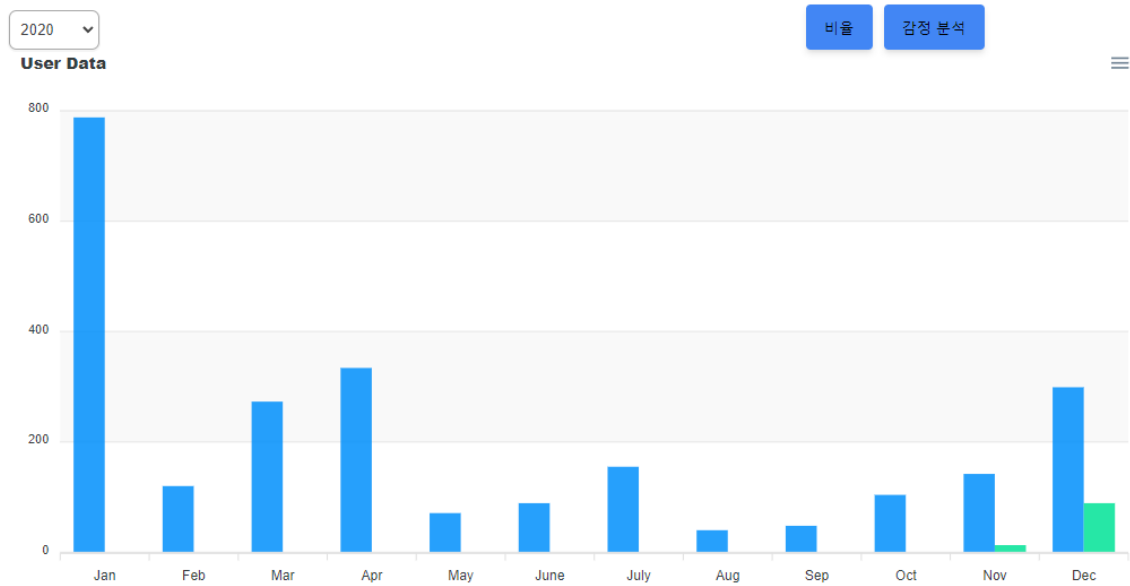
```
for (let i = 0; i < bodyJan.length; i += 900) {
  setBodyJanNum((bodyJanNum) => bodyJanNum + 1);
  axios({
    method: 'post',
    url: '/sentiment-analysis/v1/analyze',
    headers: {
      'X-NCP-APIGW-API-KEY-ID': process.env.REACT_APP_NAVER_CLIENT_ID,
      'X-NCP-APIGW-API-KEY': process.env.REACT_APP_NAVER_SECRET_KEY,
      'Content-Type': 'application/json',
      'Access-Control-Allow-Origin': '*',
    },
    data: {
      content: bodyJan.substring(i, i + 900),
    },
  })
  .then((response) => {
    setBodySentimentJanNegative(
      (bodySentimentJanNegative) =>
        (bodySentimentJanNegative +=
          response.data.document.confidence.negative)
    );
    setBodySentimentJanPositive(
      (bodySentimentJanPositive) =>
        (bodySentimentJanPositive +=
          response.data.document.confidence.positive)
    );
    setBodySentimentJanNeutral(
      (bodySentimentJanNeutral) =>
        (bodySentimentJanNeutral +=
          response.data.document.confidence.neutral)
    );
  })
  .catch((error) => {
    console.log({ error });
  });
}
```

4. 프로젝트 결과

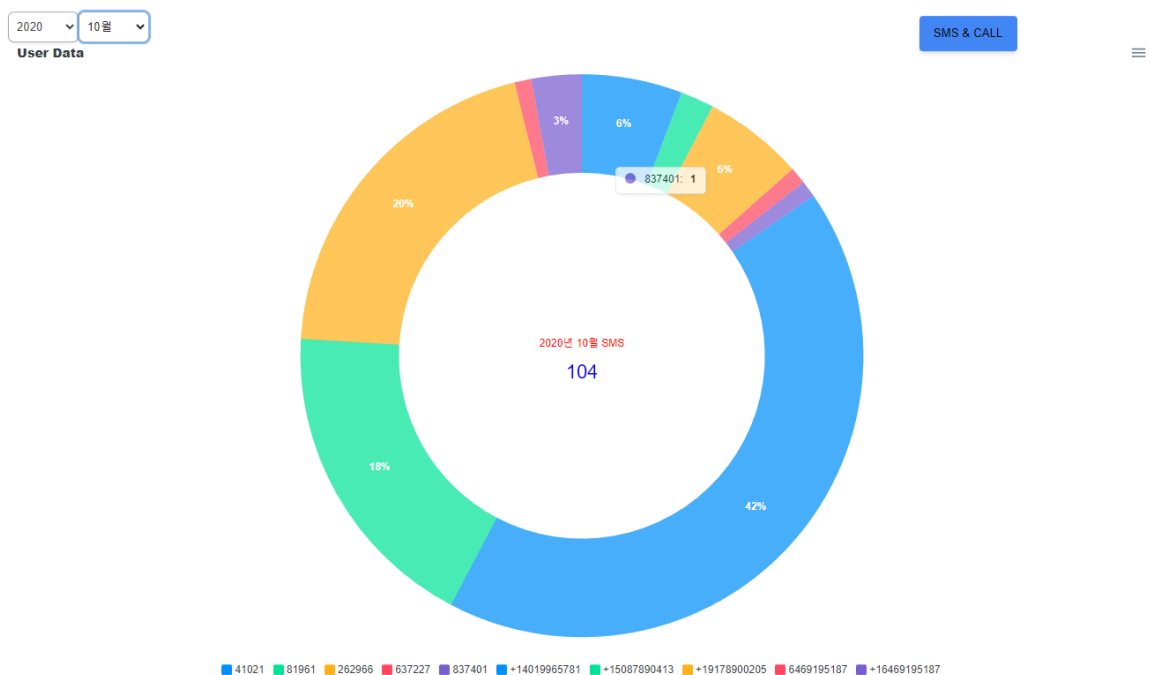
4.1 연구 결과

본 연구의 목표는 기존 연구에서 수집한 데이터를 활용하여 시각화를 하는 것이다. 이를 위해 가장 중요한 점은 시각화를 진행하기 위해 수집한 데이터를 파싱하는 것이며, 이 파싱한 데이터를 이용하여 시각화 및 연구를 진행하는 것이다.

4.1.1 문자와 전화 데이터 연도 별로 시각화



4.1.2 문자 데이터 연락처 별로 시각화



4.1.3 달 별로 전송한 문자 텍스트 감정 분석 결과 시각화

1월 부정 : 32.30261714125, 긍정 : 34.96325775, 중립 : 32.734125750000004
2월 부정 : 5.169870161, 긍정 : 35.95215773666667, 중립 : 58.877971
3월 부정 : 16.801115712599998, 긍정 : 7.942833882, 중립 : 75.256049
4월 부정 : 0.08889458044444445, 긍정 : 22.06853132944444, 중립 : 77.84257533333331
5월 부정 : 0.07829694999999999, 긍정 : 2.8508212833333333, 중립 : 97.07088333333333
6월 부정 : 14.361336013, 긍정 : 21.54468295, 중립 : 64.093981
7월 부정 : 26.6747095, 긍정 : 33.5603735, 중립 : 39.764919
8월 부정 : 34.330208, 긍정 : 16.769485, 중립 : 48.900307
9월 부정 : 0.06346868, 긍정 : 23.23921, 중립 : 76.69732
10월 부정 : 0.046157263, 긍정 : 60.492265, 중립 : 39.4615805
11월 부정 : 6.53576743775, 긍정 : 30.46215007575, 중립 : 63.002082374999999
12월 부정 : 30.508513083133337, 긍정 : 27.5433752838, 중립 : 41.948111966666666

5. 결론 및 기대효과

본 연구는 기존 “갤럭시 디지털 포렌식 도구”의 분석 및 데이터 추출에 Smart Switch 를 이용한 추출을 활용하여 데이터 파싱 및 시각화를 진행하였다. 이 시각화를 통해 사용자가 어느 연도 어느 달에 가장 많은 문자와 전화를 했는지 알 수 있었으며, 달 별로 사용자가 어떤 대상이랑 연락을 많이 했는지 알 수 있었으며, CLOVA Sentiment API 를 통해 사용자의 감정 분석까지 할 수 있었다. 이 연구를 통해 사용자가 어떤 대상이랑 연락 빈도가 높았는지 그 빈도가 높았을 때 사용자의 감정이 어땠는지를 알 수 있었다.

향후에는 루팅한 갤럭시 폰의 데이터에 대해 시각화를 진행해 볼 예정이다. 기존 연구에서 Smart Switch 로 추출한 데이터로는 루팅한 폰에서만 얻을 수 있는 데이터를 얻지 못했다. 따라서 향후에 루팅한 폰의 데이터에 대해 새로운 대상을 찾아 분석 및 파싱을 통해 시각화를 진행할 예정이다.

6. 참고문헌

- [1] <https://www.xda-developers.com/how-to-install-magisk/>
- [2] <https://namu.wiki/w/Smart%20Switch>
- [3] <https://www.ncloud.com/product/aiService/clovaSentiment>
- [4] <https://www.typescriptlang.org>
- [5] <https://ko.reactjs.org>
- [6] <https://apexcharts.com>