Esper 기반 실시간 필터링 시스템*

박세빈, 이상훈, 문양세
강원대학교 컴퓨터과학과
{sebinpark, sanghun, ysmoon}@kangwon.ac.kr

Esper-based Real-time Filtering System

Sebin Park, Sanghun Lee, Yang-Sae Moon
Dept. of Computer Science, Kangwon National University

요 약

본 논문에서는 데이터 스트림 대상의 필터링 문제를 다룬다. 데이터 스트림은 지속적으로 생성되며, 크기 또한 거대해서 이를 실시간 처리하기 위해서는 분석에 불필요한 데이터를 충분히 필터링해야 한다. 하지만, 기존 필터링 알고리즘은 하나의 데이터 형식에만 사용이 가능하여 다양하고 복잡한 스트림 환경에서는 사용하기가 어렵다. 따라서, 본 논문에서는 이 같은 문제를 해결하기 위해 스트림 형식에 따라 필터링 알고리즘을 다양하게 선택할 수 있는 필터링 시스템을 제안한다. 그리고 실시간 필터링을 위해 대표적인 오픈소스 DSMS(data stream management system)인 에스퍼 기반으로 구현한다. 또한 웹 기반 클라이언트-서버 모델로 확장 구현하여 사용자가 언제 어디에서든 필터링 시스템을 사용할 수 있게 한다. 제안하는 에스퍼 기반 실시간 필터링 시스템은 데이터 스트림으로 실시간 데이터 스트림과 벌크 데이터 스트림을 지원한다. 그리고 필터링 알고리즘으로 질의 필터링, 블룸 필터링, 베이지안 필터링을 제공한다. 제안하는 필터링 시스템 구현 결과, 데이터 스트림 특성에 적합한 필터링 알고리즘을 선택적으로 제공함으로써, 사용자가 보다 정확하고 효율적으로 의미있는 데이터를 추출 가능하게 하였다.

1. 서론

본 논문에서는 데이터 스트림 대상의 실시간 필터링 문제를 다룬다. 필터링은 스팸 메일과 같은 사용자가 원하지 않는 데이터를 제거하는 방법이다. 데이터 스트림은 실시간 생성되기 때문에 이를 처리하기 위해서는 분석에 불필요한 데이터를 충분히 제거해야 한다. 하지만, 기존 필터링 알고리즘은 대부분 특정 형식의 데이터만 필터링이 가능하다. 따라서, 다양한 형식의 데이터 스트림 또는 복합 스트림 환경에서는 분석에 불필요한 데이터 정확히 필터링하기 어렵다[1]. 본 논문에서는 이와 같은 문제를 해결하기 위해 데이터 스트림 형식과 분석 목적에 따라 필터링 알고리즘을 선택할 수 있는 필터링 시스템을 제안한다.

데이터를 처리하는 방법은 크게 배치 처리와 실시간 처리로 나뉜다. 배치 처리는 데이터를 데이터베이스에 저장 후 분석하는 방법이고, 실시간 처리는 데이터를 메인-메모리(main-memory)에서 실시간 분석하는 방법이다. 데이터 스트림은 실시간 생성되며 크기 또한 거대해서 데이터베이스에 저장하여 분석하기에는 어려움이 있다. 따라서, 데이터 스트림을 실시간 분석하기 위해서는 인-메모리(in-memory)기반의 스트림 처리 시스템(DSMS: data stream management system)[2]을 사용해야 한다. 에스퍼[3]는 데이터 스트림의 실시간 처리를 위한 대표적인 오픈소스 DSMS로, SQL과 유사하고 복합 스트림 처리가 가능한 EPL (event processing language)[4]을 지원한다. 하지만 에스퍼는 스트림 필터링에 EPL 기반의 질의 필터링만을 제공할 뿐, 기존의 다양한 필터링 알고리즘은 제공하지 않는다. 따라서, 본 논문에서는 데이터 스트림의 실시간 처리가 가능한 에스퍼에 다양한 필터링 알고리즘을 적용한 에스퍼 기반 실시간 필터링 시스템을 제안한다.

제안하는 에스퍼 기반 실시간 필터링 시스템은 클라이언트-서버 모델 기반으로 설계하고 구현한다. 클라이언트와 서버의 동작 절차는 다음과 같다. 클라이언트는 사용자가 선택한 데이터 스트림과 필터링 알고리즘, 필터 조건을 서버에게 전달한다. 그리고, 서버로부터 전달받은 필터링 결과를 실시간 보여준다. 서버는 클라이언트로부터 전달받은 정보로 필터 모델과 필터링 알고리즘을 구성하고 에스퍼에 적용하여 데이터 스트림을 실시간 필터링 한다. 그리고 필터링 결과를 클라이언트에게 실시간 반환한다.

2. 관련 연구

필터링[5]이란 데이터 신뢰도 및 분석 속도 향상을 위한 목적으로 입력 데이터에서 사용자가 원하지 않는 데이터를 제거하는 방법이다. 필터링 알고리즘은 크게 학습 기반 필터링과 비학습 기반 필터링으로 분류된다. 학습 기반 필터링은 필터링을 수행하기 전 필터 모델을 학습하여 데이터를 처리하는 방법이다. 대표적인 알고리즘은 베이지안 필터링[6], 내용 기반 필터링[1], 칼만 필터링[6] 등이 있다. 비학습 기반 필터링은 필터링 작업을 위한 별도의 학습이 필요하지 않은 방법이다. 대표적인 알고리즘은 해시 필터링[1], 질

* 이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. R7117-16-0214, 데이터 스트림 정제를 위한 지능형 샘플링 및 필터링 기술 개발).

의 필터링[7], 블룸 필터링[1] 등이 있다. 본 논문에서는 학습 기반 필터링으로 베이지안 필터링, 비학습 기반 필터링으로 질의 필터링과 블룸 필터링을 사용하여 필터링 시스템을 구현한다. 이 세 필터링은 데이터 스트림 환경에서 사용 가능한 방법으로, 분류, 탐지 검색 등의 분야에서 가장 많이 사용된다.

에스퍼는 CEP(complex event processing)[8]를 지원하는 대표적인 DSMS 오픈소스이다. 여기에서, CEP는 여러 데이터 소스로부터 발생한 대용량의 스트림을 실시간 처리하기 위한 인-메모리 기술이다. 에스퍼는 질의 수행을 위해 스트림 도메인에 특화된 언어인 EPL을 사용한다. EPL은 select, from, where, having절 등을 갖는 기존 SQL과 유사한 질의 언어이다.

그림 1은 본 논문에서 사용하는 에스퍼의 상세 구조도이다. 그림을 보면, 에스퍼는 입력 어댑터(input adapter), CEP 엔진, 출력 어댑터(output adapter)로 구성된다. 에스퍼의 자세한 설명은 참고문헌 [3]를 참조한다. 이와 같이, 에스퍼는 데이터 스트림의 실시간 처리가 가능할 뿐만 아니라 사용자가 사용하기에 용이하다. 따라서, 본 논문에서는 에스퍼에 다양한 필터링 알고리즘을 적용하여 데이터 특성과 분석 목적에 따라 필터링 알고리즘을 선택적으로 제공한다.

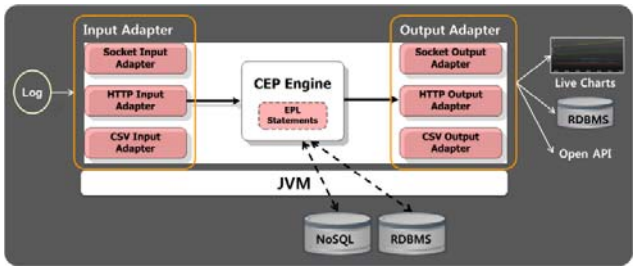

그림 1. 에스퍼 상세 구조도.

3. 에스퍼 기반 실시간 필터링 시스템

본 논문에서 제안하는 에스퍼 기반 필터링 시스템은 그림 2와 같이 클라이언트-서버 모델 기반으로 구현한다. 클라이언트-서버 구조은 대부분의 분석 및 관리 시스템에서 사용하는 네트워크 기반 모델로서, 유지보수가 용이하고 다수의 사용자가 동일한 시스템을 동시에 사용 가능한 장점이 있다. 본 논문에서는 클라이언트를 웹 기반으로 구현하여 사용자가 웹 브라우저를 통해 언제 어디서든 필터링 시스템을 사용할 수 있도록 제공한다. 클라이언트와 서버의 동작 절차는 다음과 같다. 클라이언트는 사용자로부터 데이터 스트림을 선택하고 데이터 형식과 분석 목적에 적합한 필터링 알고리즘을 선택한다. 그리고 필터링에 사용할 필터 조건을 입력받아 서버로 전송한다(①). 서버는 전달받은 필터 조건을 사용하여 데이터 스트림, 필터 모델, 필터링 알고리즘을 생성하고 이를 에스퍼에 적용한다(②). 그리고 데이터 스트림을 실시간 필터링한다(③). 필터링 된 결과는 다시 클라이언트에 실시간 전달되며, 클라이언트는 필터링 결과를 실시간 보여주고, 파일로 저장하여 제공한다(④).

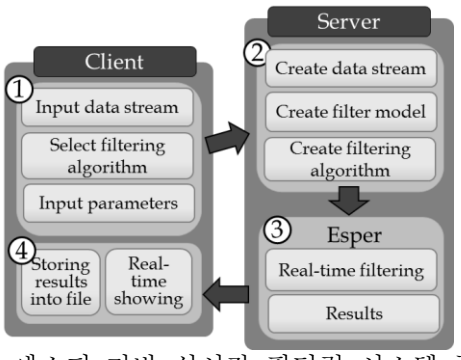

그림 2. 에스퍼 기반 실시간 필터링 시스템 구조도.

제안하는 필터링 시스템의 클라이언트는 그림 3과 같은 동작 구조를 갖는다. 그림을 보면, 클라이언트는 데이터 스트림 입력, 알고리즘 선택, 조건 입력, 통신, 출력의 다섯 가지 모듈로 구성된다. 각 모듈의 자세한 기능은 다음과 같다. **데이터 스트림 입력 모듈**: 필터링 원하는 데이터 스트림을 입력하는 모듈이다. 제안하는 필터링 시스템은 실시간 데이터 스트림과 벌크 데이터 스트림을 지원한다. 실시간 데이터 스트림은 데이터가 발생하는 소스를 입력하며 벌크 데이터 스트림은 디스크에 저장된 파일을 직접 입력한다. **알고리즘 선택 모듈**: 데이터 스트림 특성과 분석 목적에 적합한 필터링 알고리즘을 선택하는 모듈이다. 질의 필터링, 블룸 필터링, 베이지안 필터링 중 선택 가능하다. **조건 입력 모듈**: 필터 모델 생성을 위한 필터 조건을 입력하는 모듈이다. 질의 필터링은 EPL where 절에 입력할 조건을 입력하고 블룸 필터링은 블룸 필터에 등록할 조건을 입력한다. 베이지안 필터링의 경우 위의 두 필터링과는 다르게 필터 모델 생성에 학습 데이터를 사용한다. 따라서, 베이지안 필터링은 학습 데이터 생성을 위한 조건을 입력한다. **통신 모듈**: 서버와 데이터를 주고받는 모듈이다. HTTP와 웹 소켓 통신[9]을 사용하여 사용자로부터 입력받은 데이터 스트림, 필터링 알고리즘, 필터 조건을 서버에게 전달한다. 그리고 서버로부터 필터링 결과를 실시간 전달받는다. **출력 모듈**: 필터링 결과를 사용자에게 출력하는 모듈이다. 서버에서 전달받는 필터링 결과를 사용자에게 실시간 보여주며, 사용자가 저장을 요청하는 경우 필터링 결과를 파일로 저장하여 제공한다.

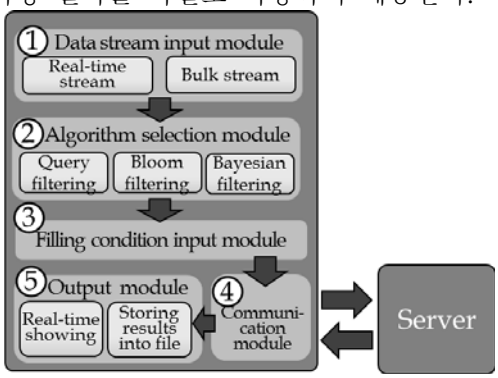

그림 3. 클라이언트 시스템 동작 구조도.

제안하는 필터링 시스템의 서버는 그림 4와 같은 동작 구조를 갖는다. 그림을 보면, 서버는 통신, 스트림 구성, 필터 생성, 알고리즘 생성, 실시간 필터링의

다섯 가지 모듈로 구성된다. 각 모듈의 자세한 기능은 다음과 같다. **통신 모듈**: 클라이언트와 데이터를 주고받는 모듈로 클라이언트의 통신 모듈과 동일하다. **스트림 구성 모듈**: 데이터 소스나 파일로부터 데이터 스트림을 구성하는 모듈이다. 클라이언트로부터 전달받은 데이터 스트림이 데이터 소스인 경우, 소스로부터 데이터를 실시간 입력받아 스트림을 구성한다. 데이터 스트림이 파일인 경우, 파일을 직접 스트림으로 변환한다. **필터 생성 모듈**: 클라이언트로부터 전달받은 필터 조건으로 필터 모델을 생성하는 모듈이다. 질의 필터링은 별도의 필터 모듈이 필요하지 않기 때문에 블룸 필터링과 베이지안 필터링에 대해 필터 모델을 생성한다. **알고리즘 생성 모듈**: 필터 모델을 사용하여 필터링 알고리즘을 생성하는 모듈이다. 질의 필터링의 경우 필터 모델이 없기 때문에, EPL의 where절에 필터 조건을 바로 적용한다. 블룸 필터링과 베이지안 필터링의 경우 앞서 생성한 필터 모델을 활용하여 알고리즘을 생성하고 이를 EPL에 적용한다. **실시간 필터링 모듈**: 필터링 알고리즘을 사용하여 데이터 스트림을 실시간 필터링하는 모듈이다. 필터링은 에스퍼 기반으로 수행하며, 필터링 결과는 통신 모듈을 통해 실시간 클라이언트에게 전달한다.

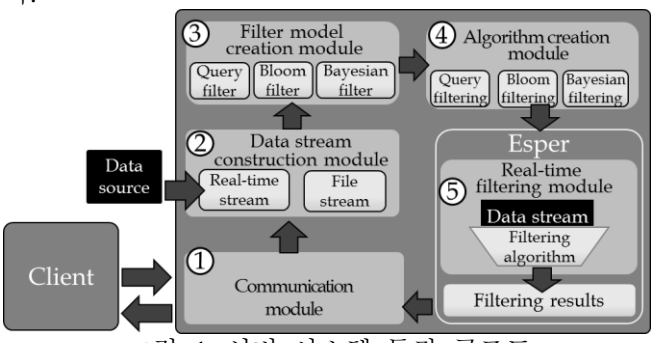

그림 4. 서버 시스템 동작 구조도.

4. 시스템 구현 및 평가

본 논문에서 제안하는 에스퍼 기반 실시간 필터링 시스템의 구현 환경은 다음과 같다. 클라이언트와 서버는 모두 Windows 8 운영체제에서 Eclipse Java EE IDE for Web Developers을 사용하여 Java와 Apache Tomcat 7.0을 기반으로 구현한다. 테스트에 사용한 실시간 스트림은 트위터 데이터로 트위터 API[10]를 통해 실시간 입력 받아 사용한다. 벌크 스트림은 트위터 데이터 3만개를 미리 저장하여 사용한다. 그림 5는 실험에 사용한 트위터 데이터의 예제이다. 그림을 보면 사용자 이름, 아이디, 생성일, 언어, 내용의 다섯 가지 스키마로 구성된 것을 알 수 있다.

{"**UserName**": "knu", "**UserID**":3157128440,
" **CreatedAt**": "Wed Aug 10 15:50:45 KST 2016",
"**Lang**": "ko", "**Text**": "@Doc_Ruby_bot 안녕하세요, 강원대학교 입니다."}

그림 5. 수집된 트위터 데이터 예제.

그림 6은 실시간 필터링 시스템의 클라이언트 초기화면이다. 그림에서 ⓐ부분은 데이터 스트림 선택 버튼으로, 실시간 스트림 또는 벌크 스트림을 선택할 수 있다. ⓑ부분은 필터링에 사용될 알고리즘을 선택하는 부분으로 블룸, 쿼리, 베이지안 필터링 중 하나를 선택 가능하다. ⓒ부분은 필터링 조건을 입력하는 부분이다.

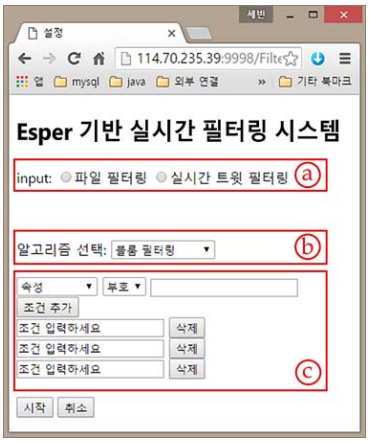

그림 6. 클라이언트 초기화면.

그림 7은 실시간 트위터 스트림을 사용한 질의 필터링 결과이다. 그림에서, ⓒ부분은 필터링 결과를 실시간 확인하는 부분이고, ⓐ부분은 필터링 된 결과를 파일로 저장하는 부분이다. 또한, ⓑ부분은 클라이언트 초기화면으로 돌아가는 부분이다. 질의 필터링에서는 필터 조건으로 'language != ko'을 사용하였다. 즉, 한국어가 아닌 트위터는 모두 필터링하였다. 그림 7의 결과 화면을 보면, 필터링 된 트위터의 언어가 모두 한국어인 것을 알 수 있다.

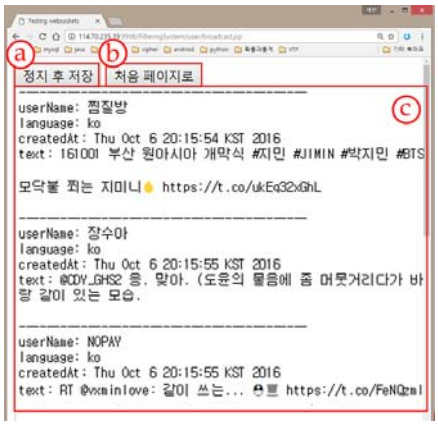

그림 7. 질의 필터링 결과.

그림 8은 실시간 트위터 스트림을 사용한 블룸 필터링 결과이다. 블룸 필터 실험에서는 필터 조건을 'language = ko'를 사용하였다. 그리고 질의 필터링과는 반대로 필터링 된 데이터를 결과로 사용하였다. 즉, 언어가 한국어인 트위터를 필터링하지 않고 결과로 추출하였다. 그림 8의 결과 화면을 보면, 그림 7의 질의 필터링 결과와 동일하게 한국어 트위터만 나타나는 것을 알 수 있다.

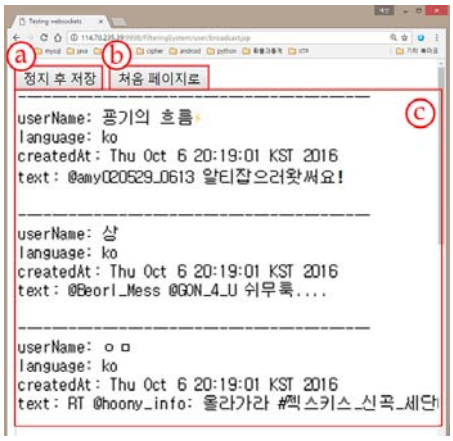

그림 8. 블룸 필터링 결과.

그림 9는 베이지안 필터링의 학습 데이터 선택 화면이다. 베이지안 실험에서는 벌크 트위터를 사용하였으며, 학습 데이터 선택을 위한 조건으로 'language = ko'를 사용하였다. 그림을 보면, 조건에 맞게 한국어 트위터만 학습 데이터로 나타난 것을 알 수 있다. 사용자는 이들 학습 데이터를 필터링해야 할 데이터와 추출해야 할 데이터로 분류한다. 즉, 왼쪽 ⓐ부분의 체크박스에 필터링해야 할 데이터를 선택하여 학습 데이터를 분류하고 서버에게 전송한다. 서버는 전달받은 학습 데이터로 필터 모델을 학습한다. 본 논문에서는 그림 9와 같이 '남자'가 나타난 트위터를 필터링 데이터로 체크하여 서버에게 전달하였다. 그림 10은 그림 9로 생성한 필터 모델로 베이지안 필터링을 수행한 결과이다. 그런데, 그림을 보면 '남자'가 포함된 트위터와 포함되지 않은 트위터 모두 필터링 결과로 나타난 것을 알 수 있다. 이는 베이지안 필터 모델의 성능이 학습 데이터 수에 크게 영향을 받기 때문이다. 즉, 본 논문에서는 적은양의 학습 데이터만 사용했기 때문에 필터링이 효율적으로 수행되지 않았다. 보다 정확한 필터링을 위해서는 대량의 학습데이터를 사용해야 한다.

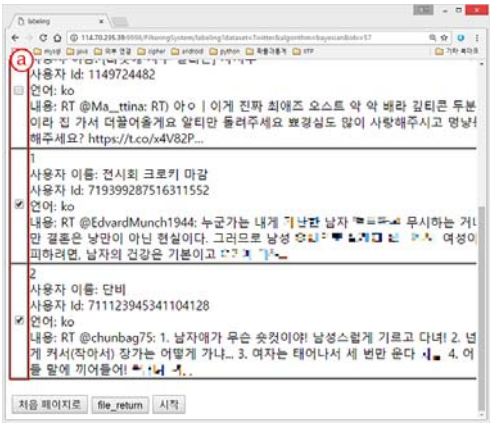

그림 9. 베이지안 필터링의 학습 데이터 선택 화면.

5. 결론

본 논문에서는 데이터 스트림 대상의 실시간 필터링 시스템을 에스퍼 기반으로 제안하고 구현하였다. 데이터 스트림은 실시간 생성되기 때문에 이를 효과적

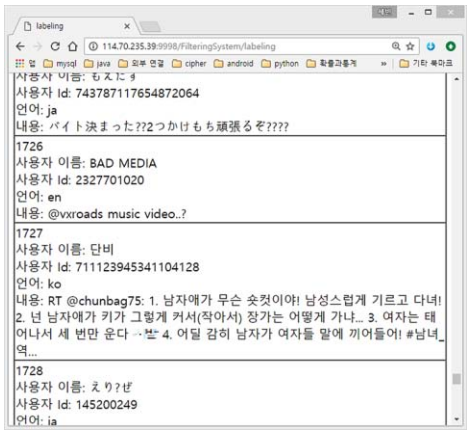

그림 10. 베이지안 필터링 결과.

으로 처리하기 위해서는 불필요한 데이터를 충분히 필터링해야 하다. 하지만, 기존 필터링 알고리즘은 특정 형식의 데이터만 필터링이 가능하여 다양하고 복잡한 데이터 스트림은 처리가 어렵다. 본 논문에서는 이 같은 문제를 해결하기 위해 스트림 형식에 적합한 필터링 알고리즘의 선택이 가능한 필터링 시스템을 제안하였다. 그리고, 데이터 스트림의 실시간 처리가 가능한 에스퍼를 기반으로 시스템을 구현하였다. 구현 결과, 제안하는 필터링 시스템은 데이터 특성에 따라 필터링 알고리즘을 선택적으로 제공함으로써, 사용자가 효율적으로 데이터 스트림을 필터링할 수 있게 하였다. 향후 연구에서는 제안하는 실시간 필터링 시스템을 분산처리가 가능한 스톰에 적용할 예정이다.

참고문헌

[1] Cambridge University Press, "Mining of Massive Datasets," *Cambridge University Press*, 2015.

[2] D. J. Abadi, et al., "Aurora: a Data Stream Management System," *ACM SIGMOD*, San Diego, California, pp. 1-18, June 2003.

[3] Esper, http://www.espertech.com.

[4] H. Li, et al., "PSTEP: A Novel Probabilistic Event Processing Language for Uncertain Spatio-Temporal Event Streams of Internet of Vehicles," *QRS* 2015, Vancouver, Canada, pp. 161-168, Aug. 2015

[5] B. D. O. Anderson and J. B. Moore, "Optimal Filtering," *Dover Publications Inc.*, 2012.

[6] E. Alpaydin, "Introduction to Machine Learning," *MitPr*, 2nd ed. 2010.

[7] B. Chazelle, "Filtering Search: A New Approach to Query-Answering," *SIAM Journal on Computing*, Vol. 15, No. 3, pp. 703-724, Aug. 1986.

[8] E. Wu, Y. Diao, and S. Rizvi, "High-Performance Complex Event Processing over Streams," *ACM SIGMOD*, Chicago, Illinois, USA, pp. 407-418, June 2006.

[9] M. R. Rahman and S. Akhter, "Real Time Bi-directional Traffic Management Support System with GPS and WebSocket," In *Proc. of the Int'l Conf. on Computer and Information Technology*, Ankra, Turkey, pp. 959-964, Oct. 2015.

[10] Twitter API, https://dev.twitter.com.