(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년03월22일
(11) 등록번호 10-2377817
(24) 등록일자 2022년03월18일

(51) 국제특허분류(Int. Cl.)
G06F 11/34 (2006.01) *G06F 11/30* (2006.01)
G06N 5/04 (2006.01) *G06N 99/00* (2019.01)
(52) CPC특허분류
G06F 11/3447 (2013.01)
G06F 11/3017 (2013.01)
(21) 출원번호 10-2017-7006514
(22) 출원일자(국제) 2015년08월28일
심사청구일자 2020년08월12일
(85) 번역문제출일자 2017년03월08일
(65) 공개번호 10-2017-0055962
(43) 공개일자 2017년05월22일
(86) 국제출원번호 PCT/US2015/047489
(87) 국제공개번호 WO 2016/040015
국제공개일자 2016년03월17일
(30) 우선권주장
14/483,800 2014년09월11일 미국(US)
(56) 선행기술조사문헌
US20130247187 A1*
(뒷면에 계속)
전체 청구항 수 : 총 26 항

(73) 특허권자
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
사라제헤 마스토레
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
천 인
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(74) 대리인
특허법인코리아나

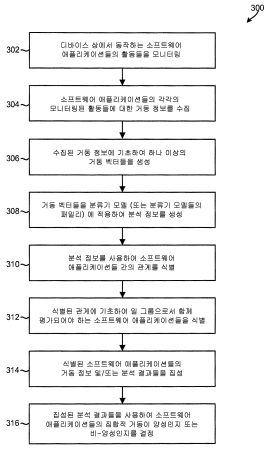
심사관 : 김계준

(54) 발명의 명칭 **모바일 디바이스 거동들에 대한 집성된 다중-애플리케이션 거동 분석을 위한 방법들 및 시스템들**

(57) 요 약

컴퓨팅 디바이스 프로세서는 디바이스 상에서 동작하는 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동을 평가하기 위해 거동 분석 및 머신 학습 기법들을 사용하는 방법들을 구현하도록 프로세서 실행가능 명령들로 구성될 수도 있다. 프로세서는 디바이스 상에서 동작하는 복수의 소프트웨어 애플리케이션들의 활동들을 모니터링하고, 모니터링된 각 활동에 대한 거동 정보를 수집하고, 수집된 거동 정보에 기초하여 거동 벡터를 생성하고, 생성된 거동 벡터를 분류기 모델에 적용하여 분석 정보를 생성하고, 분석 정보를 사용하여 복수의 소프트웨어 애플리케이션들의 집합적 거동을 분류하도록 구성될 수도 있다.

대 표 도 - 도3



(52) CPC특허분류
G06F 11/3466 (2013.01)
G06N 20/00 (2021.08)
G06N 5/04 (2013.01)
G06F 2201/865 (2013.01)

(56) 선행기술조사문헌
US20140237293 A1*
논문1(2012.12.03)*
US20140187177 A1*
US20120124422 A1*
WO2012046406 A1
*는 심사관에 의하여 인용된 문헌

명 세 서

청구범위

청구항 1

컴퓨팅 디바이스의 거동을 분석하는 방법으로서,

거동 정보를 수집하기 위해 상기 컴퓨팅 디바이스의 프로세서에서 상기 컴퓨팅 디바이스 상의 복수의 소프트웨어 애플리케이션들 간의 활동들과 상호작용들을 모니터링하는 단계;

다수의 개별 소프트웨어 애플리케이션들로부터 수집된 거동 정보를 집성하는 단계;

집성된 상기 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 집합적 거동을 특징화하는 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 단계;

다중-애플리케이션 분류기 모델에 포함된 각각의 테스트 조건을 평가하고 분석 정보를 생성하기 위해, 생성된 상기 거동 벡터 정보 구조를 상기 다중-애플리케이션 분류기 모델에 적용하는 단계로서, 상기 다중-애플리케이션 분류기 모델에서의 각각의 테스트 조건은 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들 간의 관계를 평가하는 것과 관련된 조건을 테스트하는, 상기 다중-애플리케이션 분류기 모델에 적용하는 단계;

모니터링된 상기 복수의 소프트웨어 애플리케이션들을 카테고리화하고;

상기 복수의 소프트웨어 애플리케이션들의 각각의 카테고리마다 성능 수치들을 생성하고;

상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들이 협력 작업하는지의 여부를 결정하고;

상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 평가하고; 및

평가 결과들을 생성하기 위해,

생성된 상기 분석 정보를 사용하는 단계; 및

생성된 상기 평가 결과들에 기초하여 상기 집합적 거동이 비-양성인지의 여부를 결정하는 단계를 포함하는, 컴퓨팅 디바이스의 거동을 분석하는 방법.

청구항 2

제 1 항에 있어서,

상기 집성된 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 특징화하는 상기 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 단계는, 상기 복수의 수치 값들을 통하여 상기 복수의 소프트웨어 애플리케이션들에서의 소프트웨어 애플리케이션들 모두의 상기 집합적 거동을 특징화하는 정보 구조를 생성하는 단계를 포함하는, 컴퓨팅 디바이스의 거동을 분석하는 방법.

청구항 3

제 1 항에 있어서,

상기 집성된 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 특징화하는 상기 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 단계는, 상기 복수의 수치 값들을 통하여 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들 간의 관계를 특징화하는 정보 구조를 생성하는 단계를 포함하는, 컴퓨팅 디바이스의 거동을 분석하는 방법.

청구항 4

제 1 항에 있어서,

생성된 상기 분석 정보를 사용하는 단계는, 일 그룹으로서 함께 평가되어야 하는 둘 이상의 소프트웨어 애플리케이션들을 식별하는 단계를 더 포함하는, 컴퓨팅 디바이스의 거동을 분석하는 방법.

청구항 5

제 4 항에 있어서,

부가적인 거동 정보를 수집하기 위해, 식별된 상기 둘 이상의 소프트웨어 애플리케이션들의 부가적인 활동들을 모니터링하는 단계;

수집된 상기 부가적인 거동 정보에 기초하여 상기 식별된 둘 이상의 소프트웨어 애플리케이션들의 상기 집합적 거동을 특징화하는 집합적 거동 벡터를 생성하는 단계;

부가적인 분석 정보를 생성하기 위해, 생성된 상기 집합적 거동 벡터를 상기 다중-애플리케이션 분류기 모델에 적용하는 단계; 및

상기 식별된 둘 이상의 소프트웨어 애플리케이션들의 상기 집합적 거동이 비-양성인지의 여부를 결정하기 위해 상기 부가적인 분석 정보를 사용하는 단계를 더 포함하는, 컴퓨팅 디바이스의 거동을 분석하는 방법.

청구항 6

제 4 항에 있어서,

부가적인 분석 정보를 생성하기 위해, 식별된 상기 둘 이상의 소프트웨어 애플리케이션들의 상기 거동을 각각 특징화하는 거동 벡터들을 상기 다중-애플리케이션 분류기 모델에 적용하는 단계;

상기 거동 벡터들의 각각에 대해 생성된 상기 부가적인 분석 정보를 집성하는 단계; 및

상기 식별된 둘 이상의 소프트웨어 애플리케이션들의 상기 집합적 거동이 비-양성인지의 여부를 결정하기 위해, 집성된 상기 분석 정보를 사용하는 단계를 더 포함하는, 컴퓨팅 디바이스의 거동을 분석하는 방법.

청구항 7

삭제

청구항 8

삭제

청구항 9

제 1 항에 있어서,

상기 다중-애플리케이션 분류기 모델 내의 테스트 조건들을 평가한 각각의 결과의 가중 평균치를 연산하는 단계를 더 포함하고,

생성된 상기 평가 결과들에 기초하여 상기 집합적 거동이 비-양성인지의 여부를 결정하는 단계는, 상기 집합적 거동이 상기 가중 평균치에 기초하여 비-양성인지의 여부를 결정하는 단계를 포함하는, 컴퓨팅 디바이스의 거동을 분석하는 방법.

청구항 10

제 1 항에 있어서,

생성된 상기 분석 정보를 사용하는 단계는, 상기 복수의 소프트웨어 애플리케이션들의 각각의 카테고리를 프로파일링하는 단계를 더 포함하는, 컴퓨팅 디바이스의 거동을 분석하는 방법.

청구항 11

컴퓨팅 디바이스에 있어서,

프로세서를 포함하고,

상기 프로세서는:

거동 정보를 수집하기 위해 상기 컴퓨팅 디바이스 상의 복수의 소프트웨어 애플리케이션들 간의 활동들과 상호작용들을 모니터링하고;

다수의 개별 소프트웨어 애플리케이션들로부터 수집된 거동 정보를 집성하고;

집성된 상기 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 집합적 거동을 특징화하는 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하고;

다중-애플리케이션 분류기 모델에 포함된 각각의 테스트 조건을 평가하고 분석 정보를 생성하기 위해, 생성된 상기 거동 벡터 정보 구조를 상기 다중-애플리케이션 분류기 모델에 적용하는 것으로서, 상기 다중-애플리케이션 분류기 모델에서의 각각의 테스트 조건은 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들 간의 관계를 평가하는 것과 관련된 조건을 테스트하는, 상기 다중-애플리케이션 분류기 모델에 적용하고;

모니터링된 상기 복수의 소프트웨어 애플리케이션들을 카테고리화하고;

상기 복수의 소프트웨어 애플리케이션들의 각각의 카테고리마다 성능 수치들을 생성하고;

상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들이 협력 작업하는지의 여부를 결정하고;

상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 평가하고; 그리고

평가 결과들을 생성하기 위해,

생성된 상기 분석 정보를 사용하고; 그리고

생성된 상기 평가 결과들에 기초하여 상기 집합적 거동이 비-양성인지의 여부를 결정하도록 구성되는, 컴퓨팅 디바이스.

청구항 12

제 11 항에 있어서,

상기 프로세서는, 상기 복수의 수치 값들을 통하여 상기 복수의 소프트웨어 애플리케이션들에서의 상기 소프트웨어 애플리케이션들 모두의 상기 집합적 거동을 특징화하는 정보 구조를 생성함으로써, 집성된 상기 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 특징화하는 상기 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하도록 프로세서 실행가능 명령들로 추가로 구성되는, 컴퓨팅 디바이스.

청구항 13

제 11 항에 있어서,

상기 프로세서는, 상기 복수의 수치 값들을 통하여 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들 간의 관계를 특징화하는 정보 구조를 생성함으로써, 집성된 상기 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 특징화하는 상기 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하도록 프로세서 실행가능 명령들로 추가로 구성되는, 컴퓨팅 디바이스.

청구항 14

제 11 항에 있어서,

상기 프로세서는:

생성된 상기 분석 정보를 사용하는 것이, 일 그룹으로서 함께 평가되어야 하는 둘 이상의 소프트웨어 애플리케

이션들을 식별하는 것을 추가로 포함하도록 프로세서 실행가능 명령들로 추가로 구성되는, 컴퓨팅 디바이스.

청구항 15

제 14 항에 있어서,

상기 프로세서는:

부가적인 거동 정보를 수집하기 위해, 식별된 상기 둘 이상의 소프트웨어 애플리케이션들의 부가적인 활동들을 모니터링하고;

수집된 상기 부가적인 거동 정보에 기초하여 상기 식별된 둘 이상의 소프트웨어 애플리케이션들의 상기 집합적 거동을 특징화하는 집합적 거동 벡터를 생성하고;

부가적인 분석 정보를 생성하기 위해, 생성된 상기 집합적 거동 벡터를 상기 다중-애플리케이션 분류기 모델에 적용하고; 및

상기 식별된 둘 이상의 소프트웨어 애플리케이션들의 상기 집합적 거동이 비-양성인지의 여부를 결정하기 위해 상기 부가적인 분석 정보를 사용하도록 프로세서 실행가능 명령들로 추가로 구성되는, 컴퓨팅 디바이스.

청구항 16

제 14 항에 있어서,

상기 프로세서는:

부가적인 분석 정보를 생성하기 위해, 식별된 상기 둘 이상의 소프트웨어 애플리케이션들의 거동을 각각 특징화하는 거동 벡터들을 상기 다중-애플리케이션 분류기 모델에 적용하고;

상기 거동 벡터들의 각각에 대해 생성된 상기 부가적인 분석 정보를 집성하고; 및

상기 식별된 둘 이상의 소프트웨어 애플리케이션들의 상기 집합적 거동이 비-양성인지의 여부를 결정하기 위해, 집성된 상기 분석 정보를 사용하도록 프로세서 실행가능 명령들로 추가로 구성되는, 컴퓨팅 디바이스.

청구항 17

삭제

청구항 18

삭제

청구항 19

제 11 항에 있어서,

상기 프로세서는, 상기 다중-애플리케이션 분류기 모델 내의 상기 테스트 조건들을 평가한 각각의 결과의 가중 평균치를 연산하도록 프로세서 실행가능 명령들로 추가로 구성되고; 및

상기 프로세서는, 상기 집합적 거동이 상기 가중 평균치에 기초하여 비-양성인지의 여부를 결정함으로써, 생성된 상기 평가 결과들에 기초하여 상기 집합적 거동이 비-양성인지의 여부를 결정하도록 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 20

제 11 항에 있어서,

상기 프로세서는:

상기 분석 정보를 사용하는 것이, 상기 복수의 소프트웨어 애플리케이션들의 각각의 카테고리를 프로파일링하는 것을 추가로 포함하도록 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 21

제 11 항에 있어서,

하드웨어 레벨에서 컴퓨팅 디바이스 메모리 및 하드웨어 이벤트들의 사용을 모니터링하고 수집된 거동 정보를 상기 프로세서에 출력하도록 구성된 거동 관측기 하드웨어 모듈을 더 포함하며,

상기 프로세서는, 상기 수집된 거동 정보를 상기 거동 관측기 하드웨어 모듈로부터 수신함으로써, 상기 복수의 소프트웨어 애플리케이션들의 활동들을 모니터링하도록 프로세서 실행가능 명령들로 구성되는, 컴퓨팅 디바이스.

청구항 22

저장된 프로세서 실행가능 소프트웨어 명령들을 갖는 비-일시적 컴퓨터 판독가능 저장 매체로서,

상기 저장된 프저장된 프로세서 실행가능 소프트웨어 명령들을 갖는 비-일시적 컴퓨터 판독가능 저장 매체로서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 컴퓨팅 디바이스에서의 프로세서로 하여금,

거동 정보를 수집하기 위해 상기 컴퓨팅 디바이스 상의 복수의 소프트웨어 애플리케이션들 간의 활동들과 상호작용들을 모니터링하는 것;

다수의 개별 소프트웨어 애플리케이션들로부터 수집된 거동 정보를 집성하는 것;

집성된 상기 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 집합적 거동을 특징화하는 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 것;

다중-애플리케이션 분류기 모델에 포함된 각각의 테스트 조건을 평가하고 분석 정보를 생성하기 위해, 생성된 상기 거동 벡터 정보 구조를 상기 다중-애플리케이션 분류기 모델에 적용하는 것으로서, 상기 다중-애플리케이션 분류기 모델에서의 각각의 테스트 조건은 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들 간의 관계를 평가하는 것과 관련된 조건을 테스트하는, 상기 다중-애플리케이션 분류기 모델에 적용하는 것;

모니터링된 상기 복수의 소프트웨어 애플리케이션들을 카테고리화하고;

상기 복수의 소프트웨어 애플리케이션들의 각각의 카테고리마다 성능 수치들을 생성하고;

상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들이 협력 작업하는지의 여부를 결정하고;

상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 평가하고; 및

평가 결과들을 생성하기 위해,

생성된 상기 분석 정보를 사용하는 것; 및

생성된 상기 평가 결과들에 기초하여 상기 집합적 거동이 비-양성인지의 여부를 결정하는 것을 포함하는 동작들을 수행하게 하도록 구성되는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 23

제 22 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은 프로세서로 하여금, 상기 집성된 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 특징화하는 상기 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 것이, 상기 복수의 수치 값들을 통하여 상기 복수의 소프트웨어 애플리케이션들에서의 상기 소프트웨어 애플리케이션들 모두의 상기 집합적 거동을 특징화하는 정보 구조를 생성하는 것을 포함하도록 하는 동작들을 수행하게 하도록 구성되는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 24

제 22 항에 있어서,

상기 저장된 프로세서 실행가능 명령들은 프로세서로 하여금, 상기 집성된 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 특징화하는 상기 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 것이, 상기 복수의 수치 값들을 통하여 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들 간의 관계를 특징화하는 정보 구조를 생성하는 것을 포함하도록 하는 동작들을 수행하게 하도록 구성되는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 25

제 22 항에 있어서,

상기 저장된 프로세서 실행가능 소프트웨어 명령들은:

생성된 상기 분석 정보를 사용하는 것이, 일 그룹으로서 함께 평가되어야 하는 둘 이상의 소프트웨어 애플리케이션들을 식별하는 것을 추가로 포함하도록 하는 동작들을 수행하도록 구성되는, 비-일시적 컴퓨터 판독가능 저장 매체.

청구항 26

컴퓨팅 디바이스에 있어서,

거동 정보를 수집하기 위해 상기 컴퓨팅 디바이스 상의 복수의 소프트웨어 애플리케이션들 간의 활동들과 상호작용들을 모니터링하는 수단;

다수의 개별 소프트웨어 애플리케이션들로부터 수집된 거동 정보를 집성하는 수단;

집성된 상기 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 집합적 거동을 특징화하는 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 수단;

다중-애플리케이션 분류기 모델에 포함된 각각의 테스트 조건을 평가하고 분석 정보를 생성하기 위해, 생성된 상기 거동 벡터 정보 구조를 상기 다중-애플리케이션 분류기 모델에 적용하는 수단으로서, 상기 다중-애플리케이션 분류기 모델에서의 각각의 테스트 조건은 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들 간의 관계를 평가하는 것과 관련된 조건을 각각 테스트하는, 상기 다중-애플리케이션 분류기 모델에 적용하는 수단; 및

모니터링된 상기 복수의 소프트웨어 애플리케이션들을 카테고리화하고;

상기 복수의 소프트웨어 애플리케이션들의 각각의 카테고리마다 성능 수치들을 생성하고;

상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들이 협력 작업하는지의 여부를 결정하고;

상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 평가하고; 및

평가 결과들을 생성하기 위해,

생성된 상기 분석 정보를 사용하는 수단; 및

생성된 상기 평가 결과들에 기초하여 상기 집합적 거동이 비-양성인지의 여부를 결정하는 수단을 포함하는, 컴퓨팅 디바이스.

청구항 27

제 26 항에 있어서,

상기 집성된 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 특징화하는 상기 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 수단은, 상기 복수의 수치 값들을 통하여 상기 복수의 소프트웨어 애플리케이션들에서의 상기 소프트웨어 애플리케이션들 모두의 상기 집합적 거동을 특징화하는 정보 구조를 생성하는 수단을 포함하는, 컴퓨팅 디바이스.

청구항 28

제 26 항에 있어서,

상기 집성된 거동 정보에 기초하여, 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들의 상기 집합적 거동을 특징화하는 상기 복수의 수치 값들을 포함하는 거동 벡터 정보 구조를 생성하는 수단은, 상기 복수의 수치 값들을 통하여 상기 복수의 소프트웨어 애플리케이션들 중 둘 이상의 애플리케이션들 간의 관계를 특징화하는 정보 구조를 생성하는 수단을 포함하는, 컴퓨팅 디바이스.

청구항 29

제 26 항에 있어서,

생성된 상기 분석 정보를 사용하는 수단은, 일 그룹으로서 함께 평가되어야 하는 둘 이상의 소프트웨어 애플리케이션들을 식별하는 수단을 추가로 포함하는, 컴퓨팅 디바이스.

청구항 30

제 29 항에 있어서,

부가적인 거동 정보를 수집하기 위해, 식별된 상기 둘 이상의 소프트웨어 애플리케이션들의 부가적인 활동들을 모니터링하는 수단;

수집된 상기 부가적인 거동 정보에 기초하여 상기 식별된 둘 이상의 소프트웨어 애플리케이션들의 상기 집합적 거동을 특징화하는 집합적 거동 벡터를 생성하는 수단;

부가적인 분석 정보를 생성하기 위해, 생성된 상기 집합적 거동 벡터를 상기 다중-애플리케이션 분류기 모델에 적용하는 수단; 및

상기 식별된 둘 이상의 소프트웨어 애플리케이션들의 상기 집합적 거동이 비-양성인지의 여부를 결정하기 위해 상기 부가적인 분석 정보를 사용하는 수단을 더 포함하는, 컴퓨팅 디바이스.

발명의 설명

기 술 분 야

배 경 기 술

[0001] 셀룰러 및 무선 통신 기술들은 지난 몇 년 동안 폭발적인 성장을 보여 왔다. 무선 서비스 제공자들은 이제 그들의 사용자들에게 정보, 리소스들 및 통신들로의 전례없는 레벨들의 액세스를 제공하는 다양한 피처들 (features) 과 서비스들을 제공한다. 이들 향상형태들과 보조를 맞추기 위해 소비자 전자 디바이스들 (예컨대, 셀룰러 전화들, 시계들, 헤드폰들, 리모컨 등) 은 그 어느 때보다도 강력하고 복잡해졌으며, 이제는 이들의 디바이스들 상에서 복잡하고 강력한 소프트웨어 애플리케이션들을 실행하는 것을 허용하는 강력한 프로세서들, 대용량 메모리들, 및 다른 리소스들을 공통적으로 포함한다. 이들 디바이스들은 또한 이들의 사용자들로 하여금 애플리케이션 다운로드 서비스들 (예컨대, Apple® App Store (앱 스토어), Windows® 스토어, Google® 플레이 등) 또는 인터넷으로부터 다양한 소프트웨어 애플리케이션들을 다운로드 및 실행할 수 있게 한다.

[0002] 이들 및 다른 개선들로 인해, 점점 더 많은 모바일 및 무선 디바이스 사용자들은 이제 그들의 디바이스들을 사용하여 민감한 정보 (예컨대, 신용 카드 정보, 연락처 등) 를 저장하고/하거나 보안이 중요한 작업들을 달성한다. 예를 들어, 모바일 디바이스 사용자들은 그들의 디바이스들을 자주 사용하여, 제품을 구입하고 민감한 통신들을 송신 및 수신하고, 청구서를 지불하고, 은행 계좌들을 관리하고, 다른 민감한 거래들을 수행한다. 이들 추세들로 인해, 모바일 디바이스들은 멀웨어 (malware) 및 사이버 공격들에 대한 다음 접경지대로 빠르게 되어 가고 있다.

발명의 내용

해결하려는 과제

[0003] 따라서, 모바일 및 무선 디바이스들과 같이 리소스-제약형 컴퓨팅 디바이스들을 보다 잘 보호하는, 새롭고 개선

된 보안 솔루션들이 소비자들에게 도움이 될 것이다.

과제의 해결 수단

[0004] 다양한 실시형태들은 컴퓨팅 디바이스 상에서 동작하는 2 개 이상의 소프트웨어 애플리케이션들의 집합적 거동을 평가하기 위해 거동 분석 또는 머신 학습 기법들을 사용하는 방법들을 포함한다. 본 방법들은 컴퓨팅 디바이스의 프로세서에서 복수의 소프트웨어 애플리케이션들의 활동들을 모니터링하고, 복수의 소프트웨어 애플리케이션들의 각각의 모니터링된 활동들에 대한 거동 정보를 수집하고, 수집된 거동 정보에 기초하여 거동 벡터를 생성하고, 생성된 거동 벡터를 분류기 모델에 적용하여 분석 정보를 생성하고, 분석 정보를 이용하여 복수의 소프트웨어 애플리케이션들의 집합적 거동을 평가함으로써, 컴퓨팅 디바이스에서 거동을 분석하는 것을 포함할 수도 있다.

[0005] 일 실시형태에서, 수집된 거동 정보에 기초하여 거동 벡터를 생성하는 것은, 복수의 소프트웨어 애플리케이션들의 집합적 거동을 특징화하는 정보 구조를 생성하는 것을 포함할 수도 있다. 추가적인 실시형태에서, 수집된 거동 정보에 기초하여 거동 벡터를 생성하는 것은, 복수의 소프트웨어 애플리케이션들 간의 관계를 특징화하는 정보 구조를 생성하는 것을 포함할 수도 있다. 추가적인 실시형태에서, 복수의 소프트웨어 애플리케이션들의 활동들을 모니터링하는 것은 복수의 소프트웨어 애플리케이션들 간의 상호작용들을 모니터링하는 것을 포함할 수도 있고, 분석 정보를 사용하여 복수의 소프트웨어 애플리케이션들의 집합적 거동을 평가하는 것은 일 그룹으로서 함께 평가되어야 하는 둘 이상의 소프트웨어 애플리케이션들을 식별하는 것을 포함할 수도 있다.

[0006] 추가적인 실시형태에서, 본 방법은 식별된 둘 이상의 소프트웨어 애플리케이션들의 부가적인 활동들을 모니터링하여 부가적인 거동 정보를 수집하는 것, 수집된 부가적인 거동 정보에 기초하여 그 식별된 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동을 특징화하는 집합적 거동 벡터를 생성하는 것, 생성된 집합적 거동 벡터를 분류기 모델에 적용하여 부가적인 분석 정보를 생성하는 것, 및 부가적인 분석 정보를 사용하여 그 식별된 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동이 비-양성 (non-benign) 인지의 여부를 결정하는 것을 포함할 수도 있다. 추가적인 실시형태에서, 본 방법은 식별된 둘 이상의 소프트웨어 애플리케이션들의 거동을 각각 특징화하는 거동 벡터들을 분류기 모델에 적용하여 부가적인 분석 정보를 생성하는 것, 각 거동 벡터에 대해 생성된 부가적인 분석 정보를 집성하는 것, 및 분석 결과들을 사용하여 그 식별된 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정하는 것을 포함할 수도 있다.

[0007] 추가적인 실시형태에서, 생성된 거동 벡터를 분류기 모델에 적용하여 분석 정보를 생성하는 것은 그 생성된 거동 벡터를 다중-애플리케이션 분류기 모델에 적용하는 것을 포함할 수도 있다. 추가적인 실시형태에서, 수집된 거동 정보에 기초하여 거동 벡터를 생성하는 것은 복수의 소프트웨어 애플리케이션들 중 하나의 거동을 각각 특징화하는 복수의 거동 벡터들을 생성하는 것, 및 생성된 거동 벡터를 다중-애플리케이션 분류기 모델에 적용하는 것은, 거동 벡터들의 각각을 다중-애플리케이션 분류기 모델에 적용하여 분석 정보를 생성하는 것을 포함할 수도 있다.

[0008] 추가적인 실시형태에서, 생성된 거동 벡터를 다중-애플리케이션 분류기 모델에 적용하는 것은 다중-애플리케이션 분류기 모델에 포함된 각각의 테스트 조건을 평가하는 것, 다중-애플리케이션 분류기 모델 내의 테스트 조건들을 평가한 각각의 결과의 가중 평균치를 연산하는 것, 및 집합적 거동이 가중 평균치에 기초하여 비-양성인지의 여부를 결정하는 것을 포함할 수도 있다. 추가적인 실시형태에서, 분석 정보를 사용하여 복수의 소프트웨어 애플리케이션들의 집합적 거동을 분류하는 것은, 모니터링된 복수의 소프트웨어 애플리케이션들을 카테고리화하는 것, 복수의 소프트웨어 애플리케이션들의 각 카테고리를 프로파일링하는 것, 및 복수의 소프트웨어 애플리케이션들의 각 카테고리마다 성능 수치들 (performance numbers) 을 생성하는 것을 포함할 수도 있다.

[0009] 추가적인 실시형태들은, 컴퓨팅 디바이스의 프로세서에서 복수의 소프트웨어 애플리케이션들의 활동들을 모니터링하는 것, 복수의 소프트웨어 애플리케이션들의 각각의 모니터링된 활동들에 대한 거동 정보를 수집하는 것, 수집된 거동 정보에 기초하여 거동 벡터를 생성하는 것, 생성된 거동 벡터를 분류기 모델에 적용하여 분석 정보를 생성하는 것, 및 분석 정보를 사용하여 복수의 소프트웨어 애플리케이션들의 집합적 거동을 평가하는 것을 포함하는, 동작들을 수행하도록 프로세서 실행가능 명령들로 구성되는 프로세서를 갖는 컴퓨팅 디바이스를 포함할 수도 있다.

[0010] 일 실시형태에서, 수집된 거동 정보에 기초하여 거동 벡터를 생성하는 것은 복수의 소프트웨어 애플리케이션들의 집합적 거동을 특징화하는 정보 구조를 생성하는 것을 포함할 수도 있도록 하는 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다. 추가적인 실시형태에서, 수집된 거동 정보에 기초하여

거동 벡터를 생성하는 것은 복수의 소프트웨어 애플리케이션들 간의 관계를 특징화하는 정보 구조를 생성하는 것을 포함할 수도 있도록 하는 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다.

추가적인 실시형태에서, 복수의 소프트웨어 애플리케이션들의 활동들을 모니터링하는 것은 복수의 소프트웨어 애플리케이션들 간의 상호작용들을 모니터링하는 것을 포함할 수도 있도록 하고, 분석 정보를 사용하여 복수의 소프트웨어 애플리케이션들의 집합적 거동을 평가하는 것은 일 그룹으로서 함께 평가되어야 하는 둘 이상의 소프트웨어 애플리케이션들을 식별하는 것을 포함할 수도 있도록 하는, 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다.

[0011] 추가적인 실시형태에서, 식별된 둘 이상의 소프트웨어 애플리케이션들의 부가적인 활동들을 모니터링하여 부가적인 거동 정보를 수집하는 것, 수집된 부가적인 거동 정보에 기초하여 그 식별된 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동을 특징화하는 집합적 거동 벡터를 생성하는 것, 생성된 집합적 거동 벡터를 분류기 모델에 적용하여 부가적인 분석 정보를 생성하는 것, 및 부가적인 분석 정보를 사용하여 그 식별된 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정하는 것을 더 포함하는, 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다.

[0012] 추가적인 실시형태에서, 식별된 둘 이상의 소프트웨어 애플리케이션들의 거동을 각각 특징화하는 거동 벡터들을 분류기 모델에 적용하여 부가적인 분석 정보를 생성하는 것, 각 거동 벡터에 대해 생성된 부가적인 분석 정보를 집성하는 것, 및 분석 결과들을 사용하여 그 식별된 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정하는 것을 더 포함하는, 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다. 추가적인 실시형태에서, 생성된 거동 벡터를 분류기 모델에 적용하여 분석 정보를 생성하는 것은 생성된 거동 벡터를 다중-애플리케이션 분류기 모델에 적용하는 것을 포함할 수도 있도록 하는 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다.

[0013] 추가적인 실시형태에서, 수집된 거동 정보에 기초하여 거동 벡터를 생성하는 것은 복수의 소프트웨어 애플리케이션들 중 하나의 거동을 각각 특징화하는 복수의 거동 벡터들을 생성하는 것, 및 생성된 거동 벡터를 다중-애플리케이션 분류기 모델에 적용하는 것은, 거동 벡터들의 각각을 다중-애플리케이션 분류기 모델에 적용하여 분석 정보를 생성하는 것을 포함할 수도 있도록 하는 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다.

[0014] 추가적인 실시형태에서, 생성된 거동 벡터를 다중-애플리케이션 분류기 모델에 적용하는 것은 다중-애플리케이션 분류기 모델에 포함된 각각의 테스트 조건을 평가하는 것, 다중-애플리케이션 분류기 모델 내의 테스트 조건들을 평가한 각각의 결과의 가중 평균치를 연산하는 것, 및 집합적 거동이 가중 평균치에 기초하여 비-양성인지의 여부를 결정하는 것을 더 포함할 수도 있는 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다.

[0015] 추가적인 실시형태에서, 분석 정보를 사용하여 복수의 소프트웨어 애플리케이션들의 집합적 거동을 분류하는 것은, 모니터링된 복수의 소프트웨어 애플리케이션들을 카테고리화하는 것, 복수의 소프트웨어 애플리케이션들의 각 카테고리를 프로파일링하는 것, 및 복수의 소프트웨어 애플리케이션들의 각 카테고리마다 성능 수치들을 생성하는 것을 포함할 수도 있도록 하는 동작들을 수행하도록 프로세서 실행가능 명령들로 프로세서가 구성될 수도 있다.

[0016] 추가적인 실시형태에서, 컴퓨팅 디바이스는, 하드웨어 레벨에서 컴퓨팅 디바이스 메모리 및 하드웨어 이벤트들의 사용을 모니터링하고 그 수집된 거동 정보를 프로세서에 출력하도록 구성된 거동 관측기 하드웨어 모듈을 포함할 수도 있다. 이러한 실시형태에서, 프로세서는, 복수의 소프트웨어 애플리케이션들의 모니터링 활동들이 거동 관측기 하드웨어 모듈로부터 그 수집된 거동 정보를 수신하는 것을 포함하도록 하는 동작들을 수행하도록 프로세서 실행가능 명령들로 구성될 수도 있다.

[0017] 추가적인 실시형태들은 컴퓨팅 디바이스 프로세서로 하여금 전술한 양태 방법들의 동작들을 수행하게 하도록 구성된 프로세서 실행 가능 소프트웨어 명령들이 저장된 비-일시적 컴퓨터 판독가능 저장 매체를 포함할 수도 있다. 다른 실시형태는 전술한 양태 방법들의 동작들의 기능들을 수행하기 위한 수단을 갖는 컴퓨팅 디바이스를 포함할 수도 있다.

도면의 간단한 설명

[0018] 본원에 포함되어 본 명세서의 일부를 구성하는 첨부 도면들은, 본 발명의 예시적인 양태들을 예시하며, 위에서

주어진 일반적인 설명 및 하기에 주어진 상세한 설명과 함께, 본 발명의 피처들을 설명하는 역할을 한다.

도 1 은 다양한 실시형태를 구현하기에 적합한 예시적인 시스템-온-칩 (system on chip) 의 구조도이다.

도 2 는, 특정 모바일 디바이스의 거동이 양성 (benign) 또는 비-양성인지를 결정하도록 구성되는 일 실시형태의 모바일 디바이스에서, 예시적인 논리적 컴포넌트들 및 정보 흐름들을 나타내는 블록도이다.

도 3 은 일 실시형태에 따라 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동을 평가하는 방법을 나타내는 프로세스 흐름도이다.

도 4 는 일 실시형태에 따라 둘 이상의 소프트웨어 애플리케이션들 간의 관계를 결정하는 방법을 나타내는 프로세스 흐름도이다.

도 5 는 일 실시형태에 따라 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동이 비-양성인지 여부를 결정하는 방법을 나타내는 프로세스 흐름도이다.

도 6 은 일 실시형태에 따라 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동이 비-양성인지 여부를 결정하는 방법을 나타내는 프로세스 흐름도이다.

도 7 은 모바일 디바이스에서 애플리케이션 기반 또는 희박 (lean) 분류기 모델들을 생성하는 다른 실시형태의 모바일 디바이스 방법을 나타내는 프로세스 흐름도이다.

도 8 은 일 실시형태의 서버 프로세서에 의해 생성되고 디바이스 프로세서에 의해 사용되어 희박 분류기 모델들을 생성할 수도 있는 예시적인 부스트된 판단 스텀프들 (boosted decision stumps) 의 예시이다.

도 9 는 일 실시형태에 따라 동적 및 적응 관측들을 수행하도록 구성된 관측기 모듈에서의 예시적인 논리 컴포넌트들 및 정보 흐름들을 나타내는 블록도이다.

도 10 은 다른 실시형태에 따라 관측기 데몬들 (daemons) 을 구현하는 컴퓨팅 시스템에서의 논리적 컴포넌트들 및 정보 흐름들을 나타내는 블록도이다.

도 11 은 모바일 디바이스들에 적응적인 관측을 수행하기 위한 실시형태 방법을 나타내는 프로세스 흐름도이다.

도 12 는 일 실시형태에서 사용하기에 적합한 모바일 디바이스의 컴포넌트 블록도이다.

도 13 은 일 실시형태에서 사용하기에 적합한 서버 디바이스의 컴포넌트 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0019] 첨부 도면들을 참조하면서 다양한 실시형태들을 상세히 설명하기로 한다. 가능하면 어디든지, 도면들 전반에 걸쳐 동일 또는 유사한 부분들을 지칭하기 위해 동일한 참조 번호들이 사용될 것이다. 특정 예들 및 구현형태들로 이루어진 참조들은 예시의 목적을 위한 것이며, 본 발명 또는 청구항들의 범위를 제한하도록 의도되는 것은 아니다.

[0020] 개관적으로, 다양한 실시형태들은 컴퓨팅 디바이스 상에서 동작하는 2 개 이상의 소프트웨어 애플리케이션들의 집합적 거동을 평가하기 위해 거동 분석 및 머신 학습 기법들을 사용하는 방법들 및 이 방법들을 구현하도록 구성된 컴퓨팅 디바이스들을 포함한다. 예를 들어, 일 실시형태에서, 컴퓨팅 디바이스는 디바이스 상에서 동작하는 소프트웨어 애플리케이션들의 활동들을 모니터링하고, 모니터링된 활동들로부터 거동 정보를 수집하고, 수집된 거동 정보에 기초하여 거동 벡터를 생성하고, 거동 벡터를 분석 정보를 생성하기 위한 분류기 모델에 적용하고, 분석 정보를 사용하여 소프트웨어 애플리케이션들 간의 관계를 식별하고, 식별된 관계에 기초하여 함께 평가되어야 하는 소프트웨어 애플리케이션들의 일 그룹으로서 식별하고, 식별된 소프트웨어 애플리케이션들의 분석 결과들을 집성하고, 집성된 분석 결과들을 사용하여 소프트웨어 애플리케이션들의 수집 거동이 양성인지 아니면 비-양성인지를 결정한다. 이들 동작들은 디바이스로 하여금 디바이스의 보안, 성능, 또는 전력 소비 특성들에 부정적인 영향을 미칠 수도 있고 종래의 거동 기반형 보안 솔루션들에 의해서는 달리 검출되지 못하는 다양한 조건들 또는 거동들을 신속하고 효율적으로 식별할 수 있게 하고 이에 응답할 수 있게 함으로써 컴퓨팅 디바이스의 기능을 향상시킨다.

[0021] 컴퓨팅 디바이스들에는 시간 경과에 따라 컴퓨팅 디바이스의 성능, 전력 이용 레벨들, 네트워크 사용 레벨들, 보안 및/또는 개인 정보 보호를 종종 열화시키는 (degrade) 조건들, 요인들 (factors) 및/또는 거동들을 식별, 방지 및/또는 수정하기 위한 거동 분석 기법들을 사용하는 거동 기반형 보안 시스템이 장착될 수도 있다.

예를 들어, 거동 기반형 보안 시스템은 소프트웨어 애플리케이션이 양성인지 아니면 비-양성 (예컨대, 악성, 성능-열화 등) 인지를 결정하고, 식별된 문제들 (예컨대, 비-양성으로 결정된 거동들) 을 수정하거나, 치유하거나, 치료하거나, 격리시키거나, 또는 이와 다르게 해결하기 위한 다양한 동작들을 수행하도록 구성될 수도 있다.

[0022] 이러한 거동 기반형 보안 시스템은 일반적으로 시간 경과에 따른 컴퓨팅 디바이스의 성능 열화를 방지하는 데 매우 효과적이지만, 악성 소프트웨어 애플리케이션들은 그들의 동작들을 감추기 위해 담합 (colluding) 또는 협력 작업함으로써 이러한 시스템들에 의한 검출을 피할 수도 있다. 예를 들어, 사용자의 주소록으로부터 정보를 훔칠 때 제 1 악성 소프트웨어 애플리케이션이 주소록에 액세스하여 정보를 인코딩하고 인코딩된 정보를 포괄 또는 개별 파일에 저장할 수도 있다. 그 다음 제 2 악성 애플리케이션이 포괄/개별 파일에 저장된 인코딩된 정보를 검색하여 정보를 서버에 송신할 수도 있다.

[0023] 일반적으로, 거동 기반형 보안 시스템은 이 일련의 동작들 (예컨대, 주소록 데이터를 판독, 저장, 및 송신)이 디바이스의 정상 동작 패턴들과 일치하지 않는다고 결정하고, 이 거동을 비-양성 거동으로서 분류하는 것이 가능할 것이다. 그러나, 협력 작업하는 다수의 소프트웨어 애플리케이션에 의해 동작들이 수행되므로, 기존의 솔루션들은 이들 동작들을 동일한 순서 또는 거동의 일부가 되는 것으로서 식별하지 못하는 경우가 종종 있다.

[0024] 개별적으로, 어드레스 데이터에 액세스하고, 데이터를 인코딩하고, 파일에 데이터를 저장하고, 파일에 저장된 정보를 송신하는 동작들이 반드시 비-양성 거동을 나타내는 것은 아니다. 오히려, 비-양성 거동을 나타내는 것은 이들 동작들의 집합적 또는 순차적 수행이다. 그러나, 기존의 거동 기반형 솔루션들은 소프트웨어 애플리케이션들 간의 관계들을 적절하게 특징화하지 못한다. 결과적으로, 기존의 솔루션들은 단일 거동의 일부로서 함께 평가되어야 하는 동작들을 정확하게 식별하지 못하는 경우가 종종 있다. 이러한 이유 및 다른 이유로 인해, 기존의 거동 기반형 보안 솔루션들은 다수의 소프트웨어 애플리케이션들이 협력 작업하는 사이버 공격들과 같은 소프트웨어 애플리케이션들의 일 그룹의 집합적인 활동에 의해 야기되는 거동들 및 조건들을 식별하고 이에 응답하는 데 적합하지 않다.

[0025] 기존 시스템들의 이러한 한계들을 고려하여, 다양한 실시형태들은, 담합 또는 협력 작업하는 악성 소프트웨어 애플리케이션들과 같은 소프트웨어 애플리케이션들의 일 그룹의 집합적인 활동들에 의해 야기된 비-양성 거동들을 지능적으로 및 효율적으로 식별하고 이에 응답하도록 구성된 거동 기반형 보안 시스템을 갖는 컴퓨팅 디바이스들을 구비한다.

[0026] 일 실시형태에서, 거동 기반형 보안 시스템은 소프트웨어 애플리케이션들 간의 상호작용들을 모니터링하고, 소프트웨어 애플리케이션들 간의 관계를 특징화하는 거동 벡터들을 생성하고, 거동 벡터들을 분류기 모델들에 적용하여 애플리케이션들이 담합 또는 협력 작업하고 있는지를 결정하도록 구성될 수도 있다. 그 다음, 거동 기반형 보안 시스템은 일 그룹으로서 함께 분석되어야 하는 소프트웨어 애플리케이션들을 식별하고, 식별된 애플리케이션들의 거동 벡터들을 분류기 모델에 적용하고, 결과 정보를 집성하고, 집성된 정보를 사용하여 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정할 수도 있다. 대안적으로, 거동 기반형 보안 시스템은 일 그룹으로서 함께 분석되어야 하는 소프트웨어 애플리케이션들을 식별하고, 식별된 애플리케이션들의 집합적 거동을 특징화하는 거동 벡터를 생성하고, 생성된 거동 벡터를 동일하거나 상이한 분류기 모델에 적용하여 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정할 수도 있다.

[0027] 다양한 실시형태들은 여러 가지 이유들로 컴퓨팅 디바이스 (예컨대, 모바일 컴퓨팅 디바이스) 의 기능을 향상시키고, 그 중 일부는 이하의 실시형태들의 상세한 설명들에서 설명되고 그리고/또는 이들로부터 명백해진다. 예를 들어, 담합 또는 협력 작업중인 소프트웨어 애플리케이션들을 지능적으로 식별하고 단일 디바이스 거동의 일부로서 함께 식별된 애플리케이션들의 동작들을 평가함으로써, 다양한 실시형태들은 디바이스로 하여금 종래의 거동 기반형 보안 솔루션들에 의해서는 달리 검출되지 않을 성능-열화 거동들을 식별할 수 있게 하고 이에 응답할 수 있게 함으로써, 컴퓨팅 디바이스의 기능을 향상시킨다. 또한, 다수의 개별 애플리케이션들로부터 수집된 거동 정보를 집성하고 다수의 애플리케이션들 간의 상호작용을 모니터링함으로써, 다양한 실시형태들은 디바이스로 하여금 소프트웨어 애플리케이션들 간의 관계들을 특징화할 수 있게 하고 소프트웨어 애플리케이션들의 카테고리들을 프로파일링할 수 있게 하고 소프트웨어 애플리케이션들의 일 그룹의 집합적 거동을 보다 양호하게 분석할 수 있게 하고 시스템 레벨 디바이스 거동들을 보다 양호하게 분류할 수 있게 함으로써, 컴퓨팅 디바이스의 기능을 향상시킨다.

[0028] 또한, 다양한 실시형태들은 컴퓨팅 디바이스로 하여금 컴퓨팅 디바이스의 응답성, 성능, 또는 전력 소비 특성들에 대해 현저한, 부정적인, 또는 사용자-인지가능의 영향을 미치지 않으면서 비-양성 디바이스 거동들을 신속하

고 효율적으로 식별할 수 있게 하고 이에 응답할 수 있게 하는 거동 기반형 보안 시스템을 제공한다. 이와 같이, 거동 기반형 보안 시스템은, 한정된 리소스들을 가지며 배터리 전력으로 실행되고 성능 및 보안이 중요한, 스마트폰들과 같은 모바일 디바이스들 및 다른 리소스 제약 컴퓨팅 디바이스들에 포함되거나 사용되기에 매우 적합하다.

[0029] 기능들, 기능성들, 및/또는 컴퓨팅 디바이스들의 기능에 대한 추가적인 개선점들은 이하에 제공되는 실시형태의 상세한 설명들로부터 명백해질 것이다.

[0030] "성능 열화" 라는 용어는, 이 애플리케이션에서, 처리 시간이 길고 실시간 반응 속도가 느리며 배터리 수명이 짧아지고 개인 데이터가 손실되며 악의적인 경제 활동 (예컨대, 무단 프리미엄 SMS 메시지를 송신하는 것), 서비스 거부 (DoS: denial of service), 열악하게 작성 또는 설계된 소프트웨어 애플리케이션들, 악성 소프트웨어, 멀웨어, 바이러스들, 프래그먼트화된 메모리 (fragmented memory), 스파이 (spy) 또는 봇네트 (botnet) 활동들을 위해 모바일 디바이스를 징발하거나 전화를 이용하는 것에 관련된 동작들 등과 같이 컴퓨팅 디바이스의 각종 바람직하지 않은 동작들 및 특성들을 지칭하는 데 사용된다. 또한, 이들 이유들 중 어느 하나로 인해 성능을 열화시키는 거동들, 활동들, 및 조건들은 본 명세서에서 "양성이 아님" 또는 "비-양성" 으로 지칭된다.

[0031] "모바일 컴퓨팅 디바이스" 및 "모바일 디바이스" 라는 용어는 셀룰러 전화, 스마트폰, 개인 또는 모바일 멀티미디어 플레이어, PDA (personal data assistant), 랩탑 컴퓨터, 태블릿 컴퓨터, 스마트북, 울트라북, 팜탑 컴퓨터, 무선 전자 메일 수신기, 멀티미디어 인터넷 가능형 셀룰러 전화, 무선 게임 컨트롤러, 및 메모리, 성능이 중요한 프로그래머가능형 프로세서를 포함하고 전원 절약 방법들이 혜택이 되도록 하는 배터리 전력 하에서 동작하는 유사한 개인용 전자 디바이스들 중 어느 하나 또는 전부를 본 명세서에서 상호교환가능하게 지칭하는 데 사용된다. 다양한 실시형태들은, 한정된 리소스들을 가지며 배터리로 구동되는 스마트폰들과 같은 모바일 컴퓨팅 디바이스들에 특히 유용하면서도, 그 실시형태들은 일반적으로 프로세서를 포함하고 애플리케이션 프로그램들을 실행하는 임의의 전자 디바이스에 유용하다.

[0032] 일반적으로, 모바일 디바이스의 성능 및 전력 효율성은 시간이 지남에 따라 열화된다. 최근에, 안티-바이러스 (anti-virus) 회사들 (예컨대, McAfee, Symantec 등) 은 이 열화를 더디게 할 목적으로 모바일 안티-바이러스, 방화벽, 및 암호화 제품들을 마케팅하기 시작했다. 그러나, 이 솔루션들 중의 다수는 모바일 디바이스 상에서의 연산-집약적 스캐닝 엔진 (computationally-intensive scanning engine) 의 주기적 실행에 의존하며, 이것은 모바일 디바이스의 프로세싱 및 배터리 리소스들 중의 다수를 소비할 수도 있고, 연장된 시간 주기들 동안에 모바일 디바이스를 느리게 하거나 모바일 디바이스를 쓸모없게 할 수도 있고, 그리고/또는 그렇지 않으면 사용자 경험을 열화시킬 수도 있다. 또한, 이들 솔루션들은 알려진 바이러스들 및 멀웨어를 검출하는 데에만 통상적으로 국한되며, (예컨대, 바이러스들 또는 멀웨어에 의해 성능 열화가 야기되지 않는 경우) 시간 경과에 따라 종종 결합되어 모바일 디바이스의 열화에 기여하는 여러 복잡한 요인들 및/또는 상호작용들을 해결하지는 않는다. 이러한 이유 및 다른 이유로 인해, 기존의 안티-바이러스, 방화벽, 및 암호화 제품들은, 시간의 경과에 따른 모바일 디바이스의 열화에 기여할 수도 있는 수많은 요인들을 식별하거나, 모바일 디바이스 열화를 방지하거나, 노화된 모바일 디바이스를 원래 상태로 효율적으로 복원하기 위한, 적절한 솔루션들을 제공하지는 않는다.

[0033] 또한, 현대의 모바일 디바이스들은 고도로 구성가능하고 복잡한 시스템들이다. 이와 같이, 특정 디바이스 거동이 양성인지 아니면 비-양성 (예컨대, 악성 또는 성능-열화) 인지를 결정하는 데 가장 중요한 피처들은 각 모바일 디바이스마다 다를 수도 있다. 또한, 열악하게 작성되거나 설계된 소프트웨어 애플리케이션들, 멀웨어, 바이러스들, 프래그먼트화된 메모리, 백그라운드 프로세스들 등을 포함하는, 시간의 경과에 따른 모바일 컴퓨팅 디바이스의 성능 및 전력 활용 레벨들의 열화에 기여할 수도 있는 다양한 요인들이 존재할 수도 있다. 이러한 요인들의 수, 다양성, 및 복잡성으로 인해, 현대의 모바일 컴퓨팅 디바이스들의, 복잡하지만 리소스-제약형 시스템들의 성능 및/또는 전력 활용 레벨의 열화에 기여할 수도 있는 요인들 모두를 평가하는 것은 종종 실현가능하지 않다. 이와 같이, 사용자들, 운영 체제들 및/또는 애플리케이션 프로그램들 (예컨대, 안티-바이러스 소프트웨어 등) 이 문제들의 원인들을 정확하고 효율적으로 식별하는 것은 어렵다. 결과적으로, 현재 모바일 디바이스 사용자들은 시간의 경과에 따른 모바일 디바이스의 성능 및 전력 활용 레벨의 열화를 방지하거나 노화된 모바일 디바이스들을 그 원래의 성능 및 전력 활용 레벨로 복원하는 것에 대한 방안들을 거의 가지고 있지 않다.

[0034] 현재, 컴퓨팅 디바이스 상에서 동작/실행하고 있는 소프트웨어 애플리케이션의 거동을 모델링하기 위한 다양한

솔루션들이 존재하며, 이들 솔루션들은 소프트웨어 애플리케이션이 악성인지 또는 양성인지를 결정하기 위해 머신 학습 기법들과 함께 사용될 수도 있다. 그러나, 기존의 솔루션들은 모바일 또는 리소스-제약형 디바이스들에서 사용하기에 적합하지 않은데, 왜냐하면 그들은 매우 큰 코퍼스 (corpus) 의 거동 정보를 평가할 것을 요구하고, 컴퓨팅 디바이스의 디바이스-특정형 및 애플리케이션-특정형 피처들을 동적으로 참작하기 위해 거동 모델들을 생성하지 않고, 지능적으로 거동 모델에서의 피처들을 우선순위화하지 않고, 개별적인 애플리케이션 프로그램 또는 프로세스를 평가하는 데 국한되고, 그리고/또는 디바이스에서 연산-집약적 프로세스들의 실행을 요구하기 때문이다. 이와 같이, 모바일 또는 리소스-제약형 컴퓨팅 디바이스에서 이들 솔루션을 구현하거나 수행하는 것은, 디바이스의 응답성, 성능, 또는 전력 소비 특성들에 대해 현저한, 부정적인 그리고/또는 사용자-인지가능의 영향을 미칠 수도 있다.

[0035] 이러한 문제점들을 고려하여 보다 우수한 성능을 제공하기 위해, 컴퓨팅 디바이스들 (예컨대, 모바일 디바이스들 등) 에는, 거동 분석 기법들을 사용하여 디바이스의 응답성, 성능, 또는 전력 소비 특성들에 대해 현저한, 부정적인, 또는 사용자-인지가능의 영향을 미치지 않으면서 컴퓨팅 디바이스에서의 비-양성 거동들을 지능적으로 및 효율적으로 식별하거나, 방지하거나, 수정하거나 그렇지 않으면 이에 응답하도록 구성되는, 거동 기반형 보안 시스템이 장착될 수도 있다.

[0036] 거동 기반형 보안 시스템은 관측기 프로세스, 데몬, 모듈, 또는 서브-시스템 (본 명세서에서는 통틀어서 "모듈" 이라고 지칭됨), 거동 추출기 모듈, 및 분석기 모듈을 포함할 수 있다. 관측기 모듈은 컴퓨팅 디바이스 시스템 (예컨대, 모바일 디바이스 시스템) 의 다양한 레벨들에서, 각종 애플리케이션 프로그래밍 인터페이스들 (APIs), 레지스터들, 카운터들, 또는 다른 디바이스 컴포넌트들 (본 명세서에서는, 통틀어서 "장치화된 컴포넌트들") 을 장치화 또는 조직화하고, 장치화된 컴포넌트들로부터 거동 정보를 수집하고, 수집된 거동 정보를 거동 추출기 모듈에 (예컨대, 메모리 기록 동작, 함수 호출 등을 통해) 통신하도록 구성될 수 있다. 거동 추출기 모듈은 수집된 거동 정보를 사용하여, 하나 이상의 특정 스레드들 (threads), 프로세스들, 소프트웨어 애플리케이션들, 모듈들 또는 디바이스의 컴포넌트들과 연관된, 관측된 이벤트들, 조건들, 작업들, 활동들, 및/또는 거동들 (본 명세서에서는 집합적으로 "거동들") 의 상당수 또는 모두를 각각 나타내거나 특징화하는 거동 벡터들을 생성할 수도 있다. 거동 추출기 모듈은, 생성된 거동 벡터들을 분석기 모듈에 (예컨대, 메모리 기록 동작, 함수 호출 등을 통해) 통신할 수도 있으며, 분석기 모듈은 거동 분석 동작들을 수행하기 위해 거동 벡터들을 사용하며, 데이터, 알고리즘들, 및/또는 모델들을 수행, 실행, 및/또는 적용하여 소프트웨어 애플리케이션 또는 디바이스 거동이 양성인지 또는 비-양성 (예컨대, 악성, 열악하게 작성, 성능-열화 등) 인지를 결정하는 것을 포함할 수도 있다. 그 다음, 컴퓨팅 디바이스 프로세서는 식별된 문제들 (예컨대, 비-양성이라고 결정된 거동들) 을 수정하거나, 치유하거나, 치료하거나, 격리시키거나, 또는 이와 다르게 해결하기 위한 다양한 동작들을 수행할 수도 있다.

[0037] 전술한 시스템은 일반적으로 컴퓨팅 디바이스의 성능 및 전력 활용 레벨의 열화를 방지하기 위해 매우 효과적이지만, 사이버 공격들은 점점 정교해지고 있으며, 악의적인 동작들을 숨길 수 있는 둘 이상의 소프트웨어 애플리케이션들을 사용함으로써 거동 기반형 보안 시스템에 의한 검출을 면하거나 회피할 수도 있다. 예를 들어, 두 개의 담합 소프트웨어 애플리케이션들은 그들의 동작들을 조직화하여, 사용자의 개인 정보 (예컨대, 연락처, 신용 카드 번호 등) 를 훔치고 거동 기반형 보안 시스템에 의한 검출을 피할 수도 있다. 예를 들어, 제 1 담합 애플리케이션은 디바이스의 메모리의 지정 부분 (또는 특정 메모리 위치) 에서 개인 정보를 판독 및 기록하고, 제 2 담합 애플리케이션은 메모리 위치에 저장된 정보를 판독하여 이를 서버에 송신할 수도 있다. 개별적으로, 이들은 악성 활동을 나타내지 않는 동작들이므로, 기존의 거동 기반형 보안 시스템들은 이러한 일련의 동작들을 단일 비-양성 거동과 연관된 것으로서 정확하게 식별하는 것이 가능하지 않을 수도 있다.

[0038] 일련의 동작들이 단일 거동과 연관되어 있음을 검출하고 결정하는 한 가지 방법은 데이터 흐름 추적 동작들을 수행하는 것에 의한다. FlowDroid 와 같은 데이터 흐름 추적 솔루션들은 일반적으로 악성 소프트웨어 애플리케이션들이 검출을 회피하지 못하도록 하는 데 효과적인 도구들이다. 간단히 말해, 데이터 흐름 추적 솔루션들은 컴퓨팅 시스템에서 데이터 동작들 (판독, 기록, 데이터 인코딩, 데이터 송신 등) 의 대부분 또는 전부를 모니터링하고, 개별적으로 또는 전체적으로 데이터를 부적절하게 사용하고 있는 소프트웨어 애플리케이션들을 식별하려고 시도한다. 그러나, 데이터 흐름 추적 솔루션들은 컴퓨팅 시스템에서 데이터 흐름들과 데이터 동작들의 대다수를 모니터링하는 것을 요구하고, 그리고/또는 매우 복잡하고 전력-집약적인 프로세스들의 실행을 요구한다. 이와 같이, 데이터 흐름 추적 솔루션들은 상대적으로 한정된 처리, 메모리, 및 에너지 리소스들을 갖는 통상적으로 리소스 제약형 시스템들인 모바일 디바이스들에서 사용하기에 적합하지 않다. 또한, 현대의 모바일 디바이스들은 복잡한 시스템들이며, 악의적이거나 그렇지 않으면 모바일 디바이스의 성능-열화에

기여할 수도 있는, 다양한 데이터 흐름들, 데이터 동작들 (판독, 기록, 데이터 인코딩, 데이터 송신 등), 프로세스들, 컴포넌트들, 거동들, 또는 요인들 (또는 이들의 조합들) 의 전부를 평가하는 것은 종종 실현가능하지 않다. 이러한 모든 이유로 인해, 기존의 데이터 흐름 추적 솔루션들은 모바일 및 리소스-제약형 컴퓨팅 디바이스들에서 사용하기에 적합하지 않다.

[0039] 이러한 문제점을 고려하여, 다양한 실시형태들은 디바이스에서 데이터 흐름들을 모니터링하거나 데이터 흐름 추적 동작들을 수행하지 않으면서 선택된 그룹의 소프트웨어 애플리케이션들의 집합적 거동에 의해 야기되는 담합 공격들 및 다른 조건들을 식별하고, 분석하고, 방지하고 그리고/또는 이에 응답하도록 디바이스 프로세서 (예컨대, 모바일 디바이스 프로세서 등) 를 구성할 수도 있다. 디바이스 프로세서는 컴퓨팅 디바이스의 응답성, 성능, 또는 전력 소비 특성들에 대해 현저한, 부정적인, 또는 사용자-인지가능의 영향을 미치지 않으면서 이를 달성할 수도 있다. 이와 같이, 다양한 실시형태들은, 한정된 리소스들을 포함하고 성능 및 배터리 수명이 중요한, 모바일 및 리소스-제약형 컴퓨팅 디바이스들에서 특히 유용하다.

[0040] 다양한 실시형태들에서, 디바이스 프로세서 (또는 디바이스의 거동 기반형 보안 시스템) 는 둘 이상의 소프트웨어 애플리케이션들 간의 상호작용들을 모니터링하고, 모니터링되는 애플리케이션들 간의 관계를 식별하거나 특징화하는 관계 정보 (예컨대, 거동 벡터들 등) 를 생성하고, 일 그룹으로서 함께 평가되어야 하는 소프트웨어 애플리케이션들을 식별하기 위해 관계 정보를 사용하고, 식별된 애플리케이션들의 각각으로부터 거동 정보를 수집하고, (예컨대, 거동 벡터에서의) 식별된 애플리케이션들의 각각으로부터 수집된 거동 정보를 집성하고, 그리고/또는 각각의 식별된 애플리케이션을 평가한 결과들을 (예컨대, 분류기 모델을 통해) 집성하도록 구성될 수도 있다. 그 다음, 디바이스 프로세서는, 집성된 정보를 사용하여, 식별된 애플리케이션들의 집합적 거동을 단일 디바이스 거동으로 평가할 수도 있다.

[0041] 특정 애플리케이션들 간의 관계들 및 상호작용들의 성질을 결정함으로써, 다양한 실시형태들은 디바이스 프로세서로 하여금 둘 이상의 애플리케이션들이 함께 악의적인 활동들을 숨기고 있는지의 여부 및 작거나 포커싱된 (focused) 그룹의 소프트웨어 애플리케이션들의 집합적 거동들이 (예컨대, 열악하게 설계된 애플리케이션 중 하나 이상으로 인해서 등) 컴퓨팅 디바이스의 성능 특성들에 예상치 못한 부정적인 영향을 미치는지의 여부를 보다 양호하게 결정할 수 있게 한다.

[0042] 일부 실시형태들에서, 디바이스 프로세서는 모니터링된 애플리케이션들을 카테고리화하고, 애플리케이션들의 선택 그룹들 또는 카테고리들을 프로파일링 또는 사전-프로파일링하고, 그리고/또는 애플리케이션들의 카테고리들에 대한 성능 수치들을 생성하도록 구성될 수도 있다. 성능 수치들은, 에너지 소비, 메모리 사용, 대역폭 사용, CPU 주기, 애플리케이션 성능에 대한 사용자 경험들, 사용자 인터페이스 (UI) 응답성, 및 개별 애플리케이션들, 애플리케이션들의 그룹들, 또는 애플리케이션들의 카테고리들의 다른 유사한 측정가능 특성들과 같은 다양한 성능 특성들을 식별하고, 평가하고, 그리고/또는 비교하는 데 사용하기에 적합한 정보를 포함할 수도 있다. 애플리케이션들 (또는 애플리케이션들의 그룹들 또는 카테고리들) 에 대한 성능 수치들을 프로파일링 또는 생성하기 위해 거동 분석 기법들을 사용함으로써, 다양한 실시형태들은 디바이스 프로세서로 하여금 컴퓨팅 디바이스의 성능 및/또는 전력 소비 레벨들에 대해 예상치 못한, 불균형한, 또는 부정적인 영향을 미치는 애플리케이션들 또는 애플리케이션들의 그룹들을 보다 양호하게 식별할 수 있게 하고 이에 응답할 수 있게 한다.

[0043] 다양한 실시형태들에서, 컴퓨팅 디바이스 (예컨대, 모바일 디바이스 등) 에는 본 출원에서 논의된 임의의 또는 모든 동작들을 수행하도록 구성된 포괄적인 거동 모니터링 및 분석 시스템이 구비될 수도 있다. 예를 들어, 거동 모니터링 및 분석 시스템은 관측기 모듈, 거동 추출기 모듈 및 분석기 모듈을 포함할 수도 있다. 관측기 모듈은 선택 애플리케이션들 (또는 애플리케이션의 일 그룹) 간의 동작들 (예컨대, 메모리 판독/기록 동작들), 상호작용들, 관계들 및 통신을 모니터링하도록 구성될 수도 있다. 이것은 메모리의 선택 부분들, 선택 메모리 주소들, 하드웨어 컴포넌트들, ContentResolver API 등과 같은 다양한 장치화된 컴포넌트들을 모니터링함으로써 달성될 수도 있다. 이러한 장치화된 컴포넌트들을 모니터링함으로써, 관측기 모듈은 기존의 거동 기반형 보안 시스템들에 의해서는 달리 수집되지 않을 부가적인 거동 정보를 수집할 수도 있다.

[0044] 거동 추출기 모듈은 거동 정보 (즉, 관측기 모듈에 의해 수집된 정보) 를 사용하여 애플리케이션들 간의 관계들을 특징화하는 거동 벡터들 및/또는 둘 이상의 애플리케이션들의 집합적 거동을 나타내거나 특징화하는 거동 벡터들을 생성하도록 구성될 수도 있다. 각각의 거동 벡터는 하나 이상의 "거동 피처들" 을 포함하거나 캡슐화한 정보 구조 (structure) 일 수도 있다. 거동 피처는, 컴퓨팅 디바이스에서, 관측된 이벤트, 조건, 활동, 동작, 관계, 상호작용, 또는 거동의 전부 또는 일부를 나타내는 추상적인 숫자 또는 기호일 수도 있다. 각각의 거동 피처는 가능한 값들의 범위, 이들 값들에 대해 수행될 수도 있는 동작들, 값들의 의미들, 및 다

른 유사한 정보를 식별하는 데이터 유형과 연관될 수도 있다. 데이터 유형은 컴퓨팅 디바이스에 의해 사용되어 대응 거동 피처 (또는 피처 값) 이 어떻게 측정, 분석, 가중, 또는 사용되어야 하는지를 결정할 수도 있다.

[0045] 거동 추출기 모듈은 생성된 거동 벡터들을 분석기 모듈에 (예컨대, 메모리 기록 동작, 함수 호출 등을 통해) 통신할 수도 있고, 이 분석기 모듈은 거동 벡터들을 분류기 모델들에 적용하여 소프트웨어 애플리케이션들 간의 관계들의 성질 (예컨대, 둘 이상의 소프트웨어 애플리케이션들이 협력 작업하는지의 여부 등) 을 결정할 수도 있고 그리고/또는 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정할 수도 있다.

[0046] 분류기 모델은 데이터, 엔트리들, 판단 노드들, 판단 기준들, 및/또는 특정 피처들, 요인들, 데이터 포인트들, 엔트리들, API들, 상태들, 조건들, 거동들, 소프트웨어 애플리케이션들, 프로세스들, 동작들, 컴포넌트들 등 (본 명세서에서는 통칭하여 "피처들") 을 신속하게 그리고 효율적으로 테스트 또는 평가하기 위해 디바이스 프로세서에 의해 사용될 수도 있는 정보 구조들을 포함하는 거동 모델, 또는 디바이스의 거동의 다른 실시형태들일 수도 있다. 분류기 모델은 또한 소프트웨어 애플리케이션들 간의 관계들의 성질 및/또는 컴퓨팅 디바이스에서 모니터링될 거동들을 결정하기 위해 디바이스 프로세서에 의해 사용될 수도 있는 정보를 포함할 수도 있다.

[0047] 각각의 분류기 모델은 전체 (full) 분류기 모델 또는 희박 분류기 모델로서 카테고리화될 수도 있다. 전체 분류기 모델은 수천 개의 피처들과 수십억 개의 엔트리들을 포함할 수도 있는 대형 트레이닝 (training) 데이터 세트의 함수로서 생성되는 로버스트한 (robust) 데이터 모델일 수도 있다. 희박 분류기 모델은 특정 컴퓨팅 디바이스 거동이 양성이 아닌지의 여부를 결정하는 데 가장 관련되는 피처들/엔트리들에 대한 테스트들을 포함하거나 우선순위화하는 감소된 데이터세트로부터 생성되는 보다 포커싱된 데이터 모델일 수도 있다. 로컬 분류기 모델은 모바일 컴퓨팅 디바이스에서 생성되는 희박 분류기 모델일 수도 있다. 디바이스-특정형 분류기 모델은 그 디바이스의 활동 또는 거동을 분류하는 것과 가장 관련되는 것으로 결정되는 컴퓨팅 디바이스-특정형 피처들/엔트리들만을 포함하는/테스트하는 포커싱된 데이터 모델을 포함하는 로컬 분류기 모델일 수도 있다. 애플리케이션-특정형 분류기 모델은 특정 소프트웨어 애플리케이션 (또는 특정 유형의 소프트웨어 애플리케이션) 이 비-양성인지의 여부를 결정하는 데 가장 관련되는 피처들/엔트리들에 대한 테스트들을 포함하거나 우선순위화하는 포커싱된 데이터 모델을 포함하는 로컬 분류기 모델일 수도 있다.

[0048] 다중-애플리케이션 분류기 모델은 둘 이상의 소프트웨어 애플리케이션들을 평가하는 것과 관련된 피처들을 테스트하는 집성된 피처 세트 및/또는 판단 노드들을 포함하는 로컬 분류기 모델일 수도 있다. 예를 들어, 다중-애플리케이션 분류기 모델은 두 개의 소프트웨어 애플리케이션들 간의 관계를 식별하거나 특징화하는 데 가장 관련되는 조건들 또는 피처들을 테스트하는 판단 노드들을 포함할 수 있다. 다른 예로서, 다중-애플리케이션 분류기 모델은 두 개의 소프트웨어 애플리케이션들 (또는 특정 유형의 소프트웨어 애플리케이션들) 의 집합적 거동이 비-양성인지의 여부를 결정하는 데 가장 관련되는 조건들 또는 피처들을 테스트하는 판단 노드들을 포함할 수도 있다.

[0049] 일부 실시형태들에서, 디바이스 프로세서는 둘 이상의 애플리케이션-특정형 분류기 모델들을 결합함으로써 다중-애플리케이션 분류기 모델을 생성하도록 구성될 수도 있다. 다른 실시형태들에서, 디바이스 프로세서는 둘 이상의 소프트웨어 애플리케이션들 간의 관계들, 상호작용들, 및/또는 통신을 식별하는 데 가장 관련되는 디바이스 피처들을 식별하고, 식별된 디바이스 피처들을 평가하는 테스트 조건들을 식별하고, 식별된 테스트 조건들을 포함하도록 분류기 모델을 생성함으로써, 다중-애플리케이션 분류기 모델을 생성할 수도 있다. 추가 실시형태에서, 디바이스 프로세서는 식별된 테스트 조건들의 우선순위, 중요도, 또는 성공률을 결정하고 테스트 조건들이 우선순위, 중요도, 또는 성공률에 따라 순서화되도록 분류기 모델을 생성하도록 구성될 수도 있다.

[0050] 다양한 실시형태들에서, 디바이스 프로세서는 애플리케이션들 간의 관계를 결정하기 위해 및/또는 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정하기 위해, 분류기 모델들을 생성 또는 사용하도록 구성될 수도 있다.

[0051] 예를 들어, 일 실시형태에서, 디바이스 프로세서는 컴퓨팅 디바이스 상에서 동작하는 소프트웨어 애플리케이션들 간의 상호작용들을 모니터링하고, 소프트웨어 애플리케이션들 간의 관계들을 특징화하는 거동 벡터들을 생성하고, 거동 벡터들을 분류기 모델들에 적용하여 분석 정보를 생성하고, 분석 정보를 사용하여 애플리케이션들이 담합 또는 협력 작업 중인지를 결정하도록 구성될 수도 있다. 그 다음, 디바이스 프로세서는 일 그룹으로서 함께 분석되어야 하는 소프트웨어 애플리케이션들 (예컨대, 담합한 애플리케이션들) 을 식별할 수도 있고, 식별된 애플리케이션들의 거동 벡터들을 동일 또는 상이한 분류기 모델 (또는 분류기 모델들의 패밀리 (family)) 에

적용하고, 결과 분석 정보를 집성하고, 집성된 분석 정보를 사용하여 그 식별된 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정할 수도 있다.

[0052] 다른 예로서, 디바이스 프로세서는 일 그룹으로서 함께 분석되어야 하는 소프트웨어 애플리케이션들을 식별하고, 식별된 애플리케이션들의 활동들을 모니터링하고, 모니터링된 활동들의 각각에 대한 거동 정보를 수집하고, 수집된 거동 정보에 기초하여 그 식별된 애플리케이션들의 집합적 거동을 특징화하는 거동 벡터를 생성하고, 생성된 거동 벡터를 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용하여 분석 정보를 생성하고, 분석 정보를 사용하여 그 식별된 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정하도록 구성될 수도 있다.

[0053] 일부 실시형태들에서, 디바이스 프로세서는 또한 분석 정보 (즉, 거동 벡터를 분류기 모델에 적용한 결과들) 를 사용하여 그 모니터링된 소프트웨어 애플리케이션들을 카테고리화하고, 소프트웨어 애플리케이션들의 각 카테고리를 프로파일링하고, 그리고/또는 소프트웨어 애플리케이션들의 각 카테고리마다 성능 수치들을 생성하도록 구성될 수도 있다. 예를 들어, 디바이스 프로세서는 분석 정보를 사용하여 일 클래스의 소프트웨어 애플리케이션들 (예컨대, 게임, 소셜 네트워킹, 뉴스, 금융 등) 에 의해 소비되는 전력량을 컴퓨팅/추정할 수도 있다. 또한, 각 피처의 전력 소비를 사전 프로파일링 및 측정함으로써, 디바이스 프로세서는 디바이스 상에서 동작하는 모든 활동들 또는 애플리케이션들의 전력 소비를 프로파일링할 수도 있다. 디바이스 프로세서는 이러한 정보 (예컨대, 전력 소비의 추정치들) 를 사용하여 배터리 수명을 예측하고, 디바이스의 가용 리소스들을 상당량 소비하는 애플리케이션들의 클래스 또는 클래스들을 식별하고, 다른 유사한 동작들을 수행할 수도 있다. 디바이스 프로세서는 이 정보를 컴퓨팅 디바이스의 사용자에게 디스플레이하거나 이 정보를 사용하여 디바이스 동작들을 보다 양호하게 평가할 수도 있다.

[0054] 다양한 실시형태들은 단일 프로세서 및 다중 프로세서 시스템들 및 시스템-온-칩 (SOC: system-on-chip) 을 포함하는 다수의 상이한 컴퓨팅 디바이스들에서 구현될 수도 있다. 도 1 은 다양한 실시형태들을 구현하는 컴퓨팅 디바이스들에서 사용될 수도 있는 예시적인 시스템-온-칩 (SOC) (100) 아키텍처를 나타내는 구조도이다. SOC (100) 는 디지털 신호 프로세서 (DSP) (101), 모뎀 프로세서 (104), 그래픽 프로세서 (106), 및 애플리케이션 프로세서 (108) 와 같은 다수의 이종 프로세서들을 포함할 수도 있다. SOC (100) 는 또한 하나 이상의 이종 프로세서들 (102, 104, 106, 108) 에 접속된 하나 이상의 코-프로세서들 (110) (예컨대, 벡터 코-프로세서) 을 포함할 수도 있다. 각각의 프로세서 (101, 104, 106, 108, 110) 는 하나 이상의 코어들을 포함할 수도 있고, 각 프로세서/코어는 다른 프로세서들/코어들과 독립적인 동작들을 수행할 수도 있다. 예를 들어, SOC (100) 는 제 1 유형의 운영 시스템 (예컨대, FreeBSD, LINIX, OS X 등) 및 제 2 유형의 운영 시스템 (예컨대, Microsoft Windows 8) 을 실행하는 프로세서를 포함할 수도 있다.

[0055] SOC (100) 는 또한 센서 데이터, 아날로그-디지털 변환들, 무선 데이터 송신들을 관리하고 게임들 및 영화들에 대한 인코딩된 오디오 신호들을 처리하는 것과 같은 다른 특화된 동작들을 수행하기 위한 아날로그 회로부 및 커스텀 회로부 (114) 를 포함할 수도 있다. SOC (100) 는 전압 조정기들, 발진기들, 위상 고정 루프들, 주변 브리지들, 데이터 제어기들, 메모리 제어기들, 시스템 제어기들, 액세스 포트들, 타이머들, 및 일 컴퓨팅 디바이스 상에서 실행되는 프로세서들 및 클라이언트들을 지원하는 데 사용되는 다른 유사한 컴포넌트들과 같은 시스템 컴포넌트들 및 리소스들 (116) 을 더 포함할 수도 있다.

[0056] 시스템 컴포넌트들/리소스들 (116) 및 커스텀 회로부 (114) 는 카메라들, 전자 디스플레이들, 무선 통신 디바이스들, 외부 메모리 칩들 등과 같은 주변 디바이스들과 인터페이스하기 위한 회로부를 포함할 수도 있다. 프로세서들 (101, 104, 106, 108) 은 재구성 가능한 논리 게이트들의 어레이를 포함하고/하거나 버스 아키텍처 (예컨대, CoreConnect, AMBA 등) 를 구현할 수도 있는 상호접속/버스 모듈 (124) 을 통해 하나 이상의 메모리 엘리먼트들 (112), 시스템 컴포넌트들 및 리소스들 (116) 및 커스텀 회로부 (114) 에 상호접속될 수도 있다. 통신들은 고성능 네트워크-온-칩 (NoC: networks-on chip) 과 같은 진보된 상호접속에 의해 제공될 수도 있다.

[0057] 하나 이상의 프로세서 (101, 104, 106, 108, 110) 에서 실행되는 운영 체제는 소프트웨어 애플리케이션들에 의한 메모리의 할당 및 사용을 제어 및 조직화하고, 다수의 소프트웨어 애플리케이션들에 걸쳐 물리적 메모리를 파티셔닝 (partition) 하도록 구성될 수도 있다. 이와 같이, 운영 체제는 다양한 소프트웨어 애플리케이션들에 의한 메모리의 할당 및 사용을 관리하는 하나 이상의 메모리 관리 시스템들 또는 프로세스들 (예컨대, 가상 메모리 관리기 등) 을 포함할 수도 있고, 하나의 프로세스에 의해 사용되는 메모리가 다른 프로세스에서 이미 사용중인 메모리와 간섭되지 않는 것을 확보할 수도 있다.

[0058] 앞에서 논의된 소프트웨어 기반형 메모리 관리 시스템들 또는 프로세스들 (예컨대, OS VMM 등) 에 부가하여, SOC (100) 는 중앙 처리 유닛 (CPU) 메모리 관리 유닛 (MMU) 및 시스템 MMU 와 같은 하나 이상의 하드웨어 기반형 메모리 관리 시스템들을 포함할 수도 있다. CPU MMU 및 시스템 MMU 는 가상 주소들의 물리적 주소들로의 변환, 캐시 제어, 버스 중재 (arbitration) 및 메모리 보호와 같은 다양한 메모리 관련 동작들을 수행하는 것을 담당하는 하드웨어 컴포넌트들일 수도 있다. 예를 들어, CPU MMU 는 주소 변환 서비스들 및 보호 기능성들을 메인 CPU (예컨대, 애플리케이션 프로세서 (108)) 에 제공하는 것을 담당할 수도 있으며, 시스템 MMU 는 주소 변환 서비스들 및 보호 기능성들을 다른 하드웨어 컴포넌트들 (예컨대, 디지털 신호 프로세서 (101), 모뎀 프로세서 (104), 그래픽 프로세서 (106) 등) 에 제공하는 것을 담당할 수도 있다.

[0059] SOC (100) 는 또한 하드웨어 레벨에서 그리고/또는 하드웨어 이벤트들 (예컨대, 메모리 판독 및 기록 동작들 등) 에 기초하여 소프트웨어 애플리케이션들에 의한 MMU들 및 메모리 엘리먼트들 (112) 의 액세스 또는 사용을 모니터링하도록 구성된 프로그래밍가능형 논리 회로 (PLC) 일 수도 있는 하드웨어 기반형 메모리 모니터링 유닛 (113) 을 포함할 수도 있다. 하드웨어 기반형 메모리 모니터링 유닛 (113) 은 디바이스의 다른 하드웨어 및 소프트웨어 기반형 메모리 관리 시스템들 및 MMU들과는 별개이며 독립적으로 동작할 수도 있다.

[0060] 다양한 실시형태들에서, 하드웨어 기반형 메모리 모니터링 유닛 (113) 은 메모리 사용 정보를 수집하고 수집된 메모리 사용 정보를 (PLC 에 프로그래밍될 수도 있는) 메모리 사용 패턴들과 비교하기 위해 소프트웨어 애플리케이션들에 의한 MMU들 및 메모리 엘리먼트들 (112) 의 액세스 및 사용을 모니터링하도록 구성되어, 애플리케이션들 간의 관계들을 식별하고/하거나 소프트웨어 애플리케이션들에 의한 메모리의 사용이 의심스럽거나 담합한 거동을 나타내는지의 여부를 결정할 수도 있다. 그 다음, 하드웨어 기반형 메모리 모니터링 유닛 (113) 은 식별된 관계들 및/또는 의심스러운 또는 담합한 거동들을 (예컨대, 프로세서 (101, 104, 106, 108) 를 통해) 관측기 또는 분석기 모듈들에 보고할 수도 있다.

[0061] SOC (100) 는 클록 (118) 및 전압 조정기 (120) 와 같은 SOC 외부의 리소드들과 통신하기 위한 입력/출력 모듈 (미도시) 을 더 포함할 수도 있다. SOC 외부의 리소스들 (예컨대, 클록 (118), 전압 조정기 (120)) 은 둘 이상의 내부 SOC 프로세서들/코어들 (예컨대, DSP (101), 모뎀 프로세서 (104), 그래픽 프로세서 (106), 애플리케이션 프로세서 (108) 등) 에 의해 공유될 수도 있다.

[0062] SOC (100) 는 또한 스피커들, 사용자 인터페이스 엘리먼트들 (예컨대, 입력 버튼들, 터치 스크린 디스플레이 등), 마이크로폰 어레이들, 물리적 조건들 (예컨대, 위치, 방향, 움직임, 배향, 진동, 압력 등) 을 모니터링하기 위한 센서들, 카메라들, 나침반들, GPS 수신기들, 통신 회로부 (예컨대, Bluetooth®, WLAN, WiFi 등) 및 현대의 전자 디바이스들의 다른 잘 알려진 컴포넌트들 (예컨대, 가속도계 등) 을 포함하는 센서들로부터 센서 데이터를 수집하기에 적합한 하드웨어 및/또는 소프트웨어 컴포넌트들을 포함할 수도 있다.

[0063] 위에서 논의된 SOC (100) 이외에도, 다양한 실시형태들은 단일 프로세서, 다중 프로세서들, 멀티코어 프로세서들, 또는 이들의 임의의 조합을 포함할 수도 있는 다양한 컴퓨팅 시스템들에서 구현될 수도 있다.

[0064] 도 2 는 거동 분석 기법들을 사용하여 비 양성 디바이스 거동들을 식별하고 이에 응답하도록 구성된 거동 기반형 보안 시스템 (200) 을 포함하는 일 실시형태의 모바일 컴퓨팅 디바이스 (102) 에서의 예시적인 논리 컴포넌트들 및 정보 흐름들을 나타낸다. 도 2 에 도시된 예에서, 컴퓨팅 디바이스는 거동 관측기 모듈 (202), 거동 추출기 모듈 (204), 거동 분석기 모듈 (208), 및 액츄에이터 모듈 (210) 을 포함하는 실행가능 명령 모듈들로 구성된 디바이스 프로세서 (즉, 모바일 디바이스 프로세서) 를 포함하는 모바일 컴퓨팅 디바이스 (102) 이다. 모듈들 (202-210) 의 각각은 소프트웨어, 하드웨어, 또는 이들의 조합으로 구현되는 스레드, 프로세스, 데몬, 모듈, 서브-시스템 (sub-system), 또는 컴포넌트일 수도 있다. 다양한 실시형태들에서, 모듈들 (202-210) 은 운영 체제의 일부들 내에서 (예컨대, 커널 (kernel) 내에서, 커널 공간에서, 사용자 공간에서 등), 별도의 프로그램들 또는 애플리케이션들 내에서, 특화된 하드웨어 버퍼들 또는 프로세서들에서, 또는 이들의 임의의 조합에서 구현될 수도 있다. 일 실시형태에서, 모듈들 (202-210) 중 하나 이상은 모바일 컴퓨팅 디바이스 (102) 의 하나 이상의 프로세서들 상에서 실행되는 소프트웨어 명령들로서 구현될 수도 있다.

[0065] 거동 관측기 모듈 (202) 은 디바이스의 다양한 레벨들/모듈들에서 애플리케이션 프로그래밍 인터페이스들 (APIs), 카운터들, 하드웨어 모니터들 등을 장치화하고, 활동들, 조건들, 동작들 및 이벤트들 (예컨대, 시스템 이벤트들, 상태 변경들 등) 을 일정 기간 동안 다양한 레벨들/모듈들에서 모니터링하도록 구성될 수도 있다. 예를 들어, 거동 관측기 모듈 (202) 은 모바일 컴퓨팅 디바이스 (102) 의 다양한 소프트웨어 및 하드웨어 컴포넌트들을 모니터링하고, 모바일 컴퓨팅 디바이스 (102) 의 활동들과 연관된 그 모니터링되고 측정가능한 컴포넌트들의 상호작용들, 통신들, 트랜잭션들 (transactions), 이벤트들, 또는 동작들에 관한 거동 정보를 수집하

도록 구성될 수도 있다. 이러한 활동들은 소프트웨어 애플리케이션의, 하드웨어 컴포넌트에 대한 사용, 동작 또는 작업의 수행, 소프트웨어 애플리케이션의, 모바일 컴퓨팅 디바이스 (102) 의 프로세싱 코어에서의 실행, 프로세스의 실행, 작업 또는 동작의 수행, 디바이스 거동 등을 포함한다.

[0066] 추가적인 예로서, 거동 관측기 모듈 (202) 은 소프트웨어 애플리케이션들에 의한 디바이스 메모리의 할당 또는 사용을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링하도록 구성될 수도 있다. 일 실시형태에서, 이것은 컴퓨팅 디바이스의 메모리 관리 시스템 (예컨대, 가상 메모리 관리기, 메모리 관리 유닛 등) 의 동작들을 모니터링함으로써 달성될 수도 있다. 이러한 시스템들은 일반적으로 하나의 프로세스에 의해 사용되는 메모리가 다른 프로세스에 의해 이미 사용중인 메모리와 간섭되지 않는 것을 보장하기 위해 다양한 애플리케이션 프로그램들에 의한 시스템 메모리 할당 및 사용을 관리하는 것을 담당한다. 따라서, 메모리 관리 시스템의 동작들을 모니터링함으로써, 디바이스 프로세서는, 2 개의 프로세스들이 동일한 메모리 공간을 할당받았는지, 아니면 동일한 메모리 주소 또는 위치에 정보를 판독 및 기록하고 있는지, 아니면 다른 의심스러운 메모리 관련 동작들을 수행하고 있는지와 같은, 2 개의 애플리케이션들이 협력 작업 중에 있는지를 결정할 때 사용하기에 적합한 거동 정보를 수집할 수도 있다.

[0067] 거동 관측기 모듈 (202) 은 모니터링된 활동들, 조건들, 동작들, 또는 이벤트들에 관한 거동 정보를 수집하고, 수집된 정보를 메모리에 (예컨대, 로그 파일 등에) 저장할 수도 있다. 그 다음, 거동 관측기 모듈 (202) 은 수집된 거동 정보를 거동 추출기 모듈 (204) 에 (예컨대, 메모리 기록 동작, 함수 호출 등을 통해) 통신할 수도 있다.

[0068] 일 실시형태에서, 거동 관측기 모듈 (202) 은 하드웨어 레벨에서 그리고/또는 하드웨어 이벤트들 (예컨대, 메모리 판독 및 기록 동작들 등) 에 기초하여 디바이스 메모리의 할당 또는 사용을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링하도록 구성될 수도 있다. 추가적인 실시형태에서, 거동 관측기 모듈 (202) 은 모니터링 기능들의 보다 빠르고 실시간에 가까운 실행을 위한 하드웨어 모듈 (예컨대, 도 1 을 참조하여 전술한 메모리 모니터링 유닛 (113)) 에서 구현될 수도 있다. 예를 들어, 거동 관측기 모듈 (202) 은 프로그램가능형 논리 회로 (PLC) 를 포함하는 하드웨어 모듈 내에서 구현될 수도 있으며, 여기서, 프로그램가능형 논리 엘리먼트들은 하드웨어 레벨에서 그리고/또는 하드웨어 이벤트들 (예컨대, 메모리 판독 및 기록 동작들 등) 에 기초하여 컴퓨팅 디바이스 메모리의 할당 또는 사용을 모니터링하도록 그렇지 않으면 다양한 실시형태들을 구현하도록 구성된다. 이러한 하드웨어 모듈은 거동 추출기 모듈 (204) 을 구현하는 디바이스 프로세서에 하드웨어 이벤트 모니터링의 결과들을 출력할 수도 있다. PLC 는 잘 알려진 PLC 프로그래밍 방법들을 사용하여 본 명세서에서 설명된 다양한 실시형태들의 특정 하드웨어를 모니터링하고 특정 동작들을 구현하도록 구성될 수도 있다. 하드웨어 모듈에서 실시형태 방법들의 일부 동작을 구현하기 위한 다른 회로들이 또한 사용될 수도 있다.

[0069] 유사하게, 모듈들 (202-210) 의 각각은, 이를테면 실시형태 방법들의 일부 동작을 수행하기 위해 PLC 프로그래밍 방법들을 사용하여 구성된 PLC 엘리먼트(들)를 이용하여 하나 이상의 PLC 엘리먼트들을 SoC 에 포함시킴으로써, 하드웨어 모듈들에서 구현될 수도 있다.

[0070] 거동 추출기 모듈 (204) 은 수집된 거동 정보를 수신 또는 검색하고, 이 정보를 사용하여 하나 이상의 거동 벡터들을 생성하도록 구성될 수도 있다. 다양한 실시형태들에서, 거동 추출기 모듈 (204) 은 소프트웨어 애플리케이션들에 대한 관측된 거동들, 관계들, 또는 상호작용들에 대한 간결한 정의를 포함하는 거동 벡터들을 생성하도록 구성될 수도 있다. 예를 들어, 각각의 거동 벡터는 값 또는 벡터 데이터 구조에서 소프트웨어 애플리케이션들의 집합적 거동을 간단 명료하게 설명할 수도 있다. 벡터 데이터 구조는 일련의 수치들을 포함할 수 있으며, 수치들의 각각은 컴퓨팅 디바이스의 카메라가 사용 중인지 (예컨대, 0 또는 1 로서), 컴퓨팅 디바이스로부터 송신되었거나 생성된 네트워크 트래픽이 얼마나 많은지 (예컨대, 20 KB/초 등), 통신된 인터넷 메시지들이 얼마나 많은지 (예컨대, SMS 메시지들의 수 등), 그리고/또는 거동 관측기 모듈 (202) 에 의해 수집된 임의의 다른 거동 정보와 같은, 디바이스의 피처 또는 거동을 나타낸다. 일 실시형태에서, 거동 추출기 모듈 (204) 은 거동 벡터들을 생성하도록 구성되어, 컴퓨팅 디바이스 시스템 (예컨대, 거동 분석기 모듈 (208)) 으로 하여금 애플리케이션들 간의 관계들을 신속하게 인식, 식별, 또는 분석할 수 있게 하는 식별자로서 기능할 수도 있다.

[0071] 거동 분석기 모듈 (208) 은 둘 이상의 소프트웨어 애플리케이션들 간의 관계의 성질을 식별하기 위해 거동 벡터들을 분류기 모듈들에 적용하도록 구성될 수도 있다. 거동 분석기 모듈 (208) 은 또한 집합적인 디바이스 거동 (즉, 디바이스 상에서 동작하는 둘 이상의 소프트웨어 애플리케이션들의 집합적인 활동들) 이 시간 경과에

따른 장치의 열화에 기여하고 있고 (또는 이에 기여할 가능성이 높고) 그리고/또는 이와 다르게 디바이스에 문제들을 야기할 수도 있는 비-양성 거동인지의 여부를 결정하기 위해 분류기 모듈들에 거동 벡터들을 적용하도록 구성될 수도 있다.

[0072] 거동 분석기 모듈 (208) 은 액츄에이터 모듈 (210) 에게 활동 또는 거동이 양성이 아님을 통지할 수도 있다. 이에 응답하여, 액츄에이터 모듈 (210) 은 식별된 문제들을 치유하거나, 치료하거나, 격리시키거나, 또는 이와 다르게 해결하기 위한 다양한 작동들 또는 동작들을 수행할 수도 있다. 예를 들어, 액츄에이터 모듈 (210) 은 거동 벡터를 분류기 모델에 (예컨대, 분석기 모듈에 의해) 적용한 결과가 소프트웨어 애플리케이션들의 집합적 거동이 양성이 아님을 나타내는 경우 하나 이상의 소프트웨어 애플리케이션들을 정지시키거나 종료하도록 구성될 수도 있다.

[0073] 다양한 실시형태들에서, 거동 관측기 모듈 (202) 은, 애플리케이션 프레임워크 또는 런타임 라이브러리들, 시스템 호출 API들, 파일 시스템 및 네트워킹 서브-시스템 동작들, (센서 디바이스들을 포함하는) 디바이스 상태 변경들, 및 다른 유사한 이벤트들에서 라이브러리 API 호출들에 관한 정보를 수집함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링하도록 구성될 수도 있다. 또한, 거동 관측기 모듈 (202) 은 파일 명칭들, 파일 액세스들의 카테고리들 (개인 정보 또는 일반 데이터 파일들), 파일 생성 또는 삭제 (예컨대, 유형 exe, zip 등), 파일 판독/기록/탐색 동작들, 파일 권한 변경 등에 대해 검색하는 것을 포함할 수도 있는 파일 시스템 활동을 모니터링할 수도 있다.

[0074] 거동 관측기 모듈 (202) 은 또한 접속 유형, 프로토콜, 포트 번호, 디바이스가 접속된 서버/클라이언트, 접속들의 수, 통신의 볼륨 또는 주파수를 포함할 수도 있는 데이터 네트워크 활동을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링할 수도 있다. 거동 관측기 모듈 (202) 은 송출되거나, 수신되거나, 또는 차단된 호출들 또는 메시지들 (예컨대, SMS 등) 의 유형 및 수 (예컨대, 통화된 프리미엄 호출들의 수) 를 모니터링하는 것을 포함할 수도 있는 전화 네트워크 활동을 모니터링할 수도 있다.

[0075] 거동 관측기 모듈 (202) 은 또한 포크들 (forks) 의 수, 메모리 액세스 동작들, 열린 파일들의 수 등을 모니터링하는 것을 포함할 수도 있는 시스템 리소스 사용을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링할 수도 있다. 거동 관측기 모듈 (202) 은, 디스플레이가 켜져 있는지 또는 꺼져 있는지, 디바이스가 잠금되어 있는지 또는 잠금해제되어 있는지, 배터리의 잔량, 카메라의 상태 등과 같은 각종 요인들을 모니터링하는 것을 포함할 수도 있는, 모바일 컴퓨팅 디바이스 (102) 의 상태를 모니터링할 수도 있다. 거동 관측기 모듈 (202) 은 또한, 예를 들어, 중대한 서비스들 (브라우저, 계약들 제공자 등), 프로세스 간 통신의 정도, 팝-업 (pop-up) 윈도우들 등에 대한 인텐트들을 모니터링함으로써 프로세스 간 통신 (IPC: inter-process communication) 을 모니터링할 수도 있다.

[0076] 거동 관측기 모듈 (202) 은 또한 카메라들, 센서들, 전자 디스플레이들, WiFi 통신 컴포넌트들, 데이터 제어기들, 시스템 제어기들, 액세스 포트들, 타이머들, 주변 디바이스들, 무선 통신 컴포넌트들, 외부 메모리 칩들, 전압 조정기들, 발진기들, 위상 고정 루프들, 주변 브리지들, 및 모바일 컴퓨팅 디바이스 (102) 상에서 구동되는 프로세서들 및 클라이언트들을 지원하는 데 사용되는 다른 유사한 컴포넌트들을 포함할 수도 있는 하나 이상의 하드웨어 컴포넌트들의 드라이버 통계 및/또는 스테이터스 (status) 를 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링할 수도 있다.

[0077] 거동 관측기 모듈 (202) 은 또한 모바일 컴퓨팅 디바이스 (102) 및/또는 컴퓨팅 디바이스 서브-시스템들의 상태 또는 스테이터스를 나타내는 하나 이상의 하드웨어 카운터들을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링할 수도 있다. 하드웨어 카운터는 모바일 컴퓨팅 디바이스 (102) 에서 발생하는 하드웨어 관련 활동들 또는 이벤트들의 카운트 값 또는 상태를 저장하도록 구성된 프로세서들/코어들의 특수-목적 레지스터를 포함할 수도 있다.

[0078] 거동 관측기 모듈 (202) 은 또한 소프트웨어 애플리케이션들의 작동들 또는 동작들, 애플리케이션 다운로드 서버 (예컨대, Apple® App Store 서버) 로부터의 소프트웨어 다운로드, 소프트웨어 애플리케이션들에 의해 사용되는 컴퓨팅 디바이스 정보, 호출 정보, 텍스트 메시징 정보 (예컨대, SendSMS, BlockSMS, ReadSMS 등), 미디어 메시징 정보 (예컨대, ReceiveMMS), 사용자 계정 정보, 위치 정보, 카메라 정보, 가속도계 정보, 브라우저 정보, 브라우저 기반형 통신의 콘텐츠, 음성 기반형 통신의 콘텐츠, 단거리 라디오 통신 (예컨대, Bluetooth, WiFi 등), 텍스트 기반형 통신들의 콘텐츠, 레코딩된 오디오 파일들의 콘텐츠, 전화번호부 또는 연락처 정보, 연락처 리스트들 등을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링할 수도 있다.

[0079] 거동 관측기 모듈 (202) 은 또한 음성메일 (VoiceMailComm), 디바이스 식별자들 (DeviceIDComm), 사용자 계정 정보 (UserAccountComm), 달력 정보 (CalendarComm), 위치 정보 (LocationComm), 레코딩된 오디오 정보 (RecordAudioComm), 가속도계 정보 (AccelerometerComm) 등을 포함하는 통신들을 포함하는, 모바일 컴퓨팅 디바이스 (102) 의 송신들 또는 통신들을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링할 수도 있다.

[0080] 거동 관측기 모듈 (202) 은 또한 나침반 정보, 컴퓨팅 디바이스 설정들, 배터리 수명, 자이로스코프 정보, 압력 센서들, 자석 센서들, 스크린 활동 등의 사용, 및 업데이트/변경을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 의 활동들을 모니터링할 수도 있다. 거동 관측기 모듈 (202) 은 소프트웨어 애플리케이션으로 그리고 이로부터 통신된 통지들 (AppNotifications), 애플리케이션 업데이트들 등을 모니터링할 수도 있다. 거동 관측기 모듈 (202) 은 제 2 소프트웨어 애플리케이션의 다운로드 및/또는 설치를 요청하는 제 1 소프트웨어 애플리케이션에 관한 조건들 또는 이벤트들을 모니터링할 수도 있다. 거동 관측기 모듈 (202) 은 패스워드의 입력 등과 같은 사용자 검증에 관한 조건들 또는 이벤트들을 모니터링할 수도 있다.

[0081] 거동 관측기 모듈 (202) 은 또한 애플리케이션 레벨, 라디오 레벨, 및 센서 레벨을 포함하는 모바일 컴퓨팅 디바이스 (102) 의 다수의 레벨들에서 조건들 또는 이벤트들을 모니터링함으로써 모바일 컴퓨팅 디바이스 (102) 활동들을 모니터링할 수도 있다. 애플리케이션 레벨 관측들은 안면 인식 소프트웨어를 통해 사용자를 관측하는 것, 소셜 스트림들 (social streams) 을 관측하는 것, 사용자에 의해 입력된 메모들을 관측하는 것, PassBook®, Google® 지갑, Paypal®, 및 다른 유사한 애플리케이션들 또는 서비스들의 사용에 관한 이벤트들을 관측하는 것을 포함할 수도 있다. 애플리케이션 레벨 관측들은 또한 가상 사설 네트워크들 (VPN: virtual private networks) 의 사용에 관한 이벤트들 및 동기화, 음성 검색들, 음성 제어 (예컨대, 하나의 단어를 말함으로써 전화를 잠금/잠금해제), 언어 번역기들, 연산들을 위한 데이터의 오프로딩 (offloading), 비디오 스트리밍, 사용자 활동 없는 카메라 사용, 사용자 활동 없는 마이크로폰 사용 등에 관한 이벤트들을 관측하는 것을 포함할 수도 있다.

[0082] 라디오 레벨 관측들은 라디오 통신 링크들을 확립하거나 정보를 송신하기 전의 모바일 컴퓨팅 디바이스 (102) 와의 사용자 상호작용, 듀얼/다중 가입자 식별 모듈 (SIM) 카드들, 인터넷 라디오, 모바일 전화 테더링 (tethering), 연산들을 위한 오프로딩 데이터, 디바이스 상태 통신들, 게임 제어기 또는 홈 제어기로서의 사용, 차량 통신들, 컴퓨팅 디바이스 동기화 등 중 어느 하나 이상의 현존, 존재 또는 양을 결정하는 것을 포함할 수도 있다. 라디오 레벨 관측들은 또한 위치확인, 피어-투-피어 (p2p: peer-to-peer) 통신들, 동기화, 차량 대 차량 통신들, 및/또는 머신-대-머신 (m2m: machine-to-machine) 을 위한 라디오들 (WiFi, WiMax, Bluetooth 등) 의 사용을 모니터링하는 것을 포함할 수도 있다. 라디오 레벨 관측들은 네트워크 트래픽 사용, 통계, 또는 프로파일들을 모니터링하는 것을 더 포함할 수도 있다.

[0083] 센서 레벨 관측들은 모바일 컴퓨팅 디바이스 (102) 의 사용 및/또는 외부 환경을 결정하기 위해 자석 센서 또는 다른 센서를 모니터링하는 것을 포함할 수도 있다. 예를 들어, 컴퓨팅 디바이스 프로세서는 디바이스가 (예컨대, 수납주머니 (holster) 내의 자석을 감지하도록 구성된 자석 센서를 통해) 수납주머니 내에 있는지, 또는 (예컨대, 카메라 또는 광 센서에 의해 검출된 광의 양을 통해) 사용자의 주머니 내에 있는지 결정하도록 구성될 수도 있다. 모바일 컴퓨팅 디바이스 (102) 가 수납주머니 내에 있음을 검출하는 것은 의심스러운 거동들을 인식하는 것과 관련될 수도 있는데, 이는, 예를 들어, 모바일 컴퓨팅 디바이스 (102) 가 수납되는 동안에 발생하는 사용자에 의한 능동적인 사용 (예컨대, 사진들 또는 비디오들의 촬영, 메시지들의 송신, 음성 호출을 수행, 사운드들을 레코딩 등) 에 관한 활동들 및 기능들이 (예컨대, 사용자를 추적하거나 정찰하기 위해) 디바이스 상에서 실행되는 범죄적 프로세스들의 징후들일 수도 있기 때문이다.

[0084] 사용 또는 외부 환경들과 관련된 센서 레벨 관측들에 대한 다른 예들은, NFC 시그널링을 검출하는 것, 신용카드 스캐너, 바코드 스캐너, 또는 모바일 태그 판독기로부터 정보를 수집하는 것, 범용 직렬 버스 (USB: Universal Serial Bus) 전력 충전 소스의 존재를 검출하는 것, 키보드 또는 보조 디바이스가 모바일 컴퓨팅 디바이스 (102) 에 커플링되었음을 검출하는 것, 모바일 컴퓨팅 디바이스 (102) 가 (예컨대, USB 등을 통해) 다른 컴퓨팅 디바이스에 커플링되었음을 검출하는 것, LED, 플래시, 플래시광, 또는 광원이 수정되었거나 사용불가능하게 되었는지의 여부 (예컨대, 긴급 시그널링 앱을 악의적으로 사용불가능하게 하는 것 등) 를 결정하는 것, 스피커 또는 마이크로폰이 켜지거나 전원이 인가되었음을 검출하는 것, 충전 또는 전력 이벤트를 검출하는 것, 모바일 컴퓨팅 디바이스 (102) 가 게임 컨트롤러로서 사용되고 있음을 검출하는 것 등을 포함할 수도 있다. 센서 레벨 관측들은 또한 의료 또는 건강관리 센서들로부터 또는 사용자의 신체를 스캔한 것으로부터 정보를 수집하는 것, USB/오디오 잭에 연결된 외부 센서로부터 정보를 수집하는 것, 촉각 또는 햅틱 센서로부터 정보를 (예컨

대, 진동자 인터페이스 등을 통해) 수집하는 것, 모바일 컴퓨팅 디바이스 (102) 의 열 (thermal) 상태에 관한 정보를 수집하는 것 등을 포함할 수도 있다.

[0085] 일 실시형태에서, 모니터링되는 요인들의 수를 관리가능한 레벨로 감소시키기 위해, 거동 관측기 모듈 (202) 은 컴퓨팅 디바이스의 열화에 기여할 수도 있는 모든 요인들의 작은 서브세트인 거동들 또는 요인들의 초기 세트를 모니터링/관측함으로써 대략적 (coarse) 관측들을 수행하도록 구성될 수도 있다. 일 실시형태에서, 거동 관측기 모듈 (202) 은 서버 및/또는 클라우드 서비스 또는 네트워크에서의 컴포넌트로부터 거동들 및/또는 요인들의 초기 세트를 수신할 수도 있다. 일 실시형태에서, 거동들/요인들의 초기 세트는 머신 학습 분류기 모델들에서 특정될 수도 있다.

[0086] 각각의 분류기 모델은 컴퓨팅 디바이스의 거동의 특정 피처 또는 실시형태를 평가하기 위해 컴퓨팅 디바이스 프로세서에 의해 사용될 수도 있는 데이터 및/또는 정보 구조들 (예컨대, 피처 벡터들, 거동 벡터들, 컴포넌트 리스트들 등) 을 포함하는 거동 모델일 수도 있다. 각각의 분류기 모델은 또한 컴퓨팅 디바이스에서 다수의 피처들, 요인들, 데이터 포인트들, 엔트리들, API들, 상태들, 조건들, 거동들, 애플리케이션들, 프로세스들, 동작들, 컴포넌트들 등 (본 명세서에서는 통칭하여 "피처들") 을 모니터링하기 위한 판단 기준들을 포함할 수 있다. 분류기 모델들은 컴퓨팅 디바이스 상에 사전-설치되거나, 네트워크 서버로부터 다운로드 또는 수신되거나, 컴퓨팅 디바이스에서 생성되거나, 또는 이들의 임의의 조합일 수도 있다. 분류기 모델들은 크라우드 소싱 (crowd sourcing) 솔루션들, 거동 모델링 기법들, 머신 학습 알고리즘들 등을 사용함으로써 생성될 수도 있다.

[0087] 각각의 분류기 모델은 전체 분류기 모델 또는 희박 분류기 모델로서 카테고리화될 수도 있다. 전체 분류기 모델은 수천 개의 피처들과 수십억 개의 엔트리들을 포함할 수도 있는 대형 트레이닝 (training) 데이터세트의 함수로서 생성되는 로버스트한 데이터 모델일 수도 있다. 희박 분류기 모델은 특정 활동이 진행중인 중요한 활동인지의 여부 및/또는 특정 컴퓨팅 디바이스 거동이 양성이 아닌지의 여부를 결정하는 데 가장 관련되는 피처들/엔트리들만을 포함/테스트하는 감소된 데이터세트로부터 생성되는 보다 포커싱된 데이터 모델일 수도 있다. 일례로서, 디바이스 프로세서는 네트워크 서버로부터 전체 분류기 모델을 수신하고, 전체 분류기에 기초하여 컴퓨팅 디바이스에서 희박 분류기 모델을 생성하고, 국소적으로 생성된 희박 분류기 모델을 사용하여 디바이스의 거동을 양성 또는 비-양성 (즉, 악성, 성능-열화 등) 중 어느 일방인 것으로 분류하도록 구성될 수도 있다.

[0088] 국소적으로 생성된 희박 분류기 모델은 컴퓨팅 디바이스에서 생성되는 희박 분류기 모델이다. 즉, 현대의 컴퓨팅 디바이스들 (예컨대, 모바일 디바이스들 등) 은 고도로 구성가능하고 복잡한 시스템들이므로, 특정 디바이스 거동이 비-양성인지 (예컨대, 악성 또는 성능-열화) 의 여부를 결정하는 데 가장 중요한 피처들은 각 디바이스마다 상이할 수도 있다. 또한, 피처들의 상이한 조합은 특정 거동이 비-양성인지의 여부를 그 디바이스가 신속히 그리고 효율적으로 결정하기 위해 각 디바이스에서의 모니터링 및/또는 분석을 요구할 수도 있다. 그러나, 모니터링 및 분석을 요구하는 피처들의 정확한 조합, 및 각 피처 또는 피처 조합의 상대적 우선순위 또는 중요도는, 종종, 오직, 거동이 모니터링 또는 분석되어야 하는 특정 디바이스로부터 획득된 정보를 사용하여 결정될 수 있다. 이러한 이유 및 다른 이유로 인해, 다양한 실시형태들은 모델들이 사용되는 컴퓨팅 디바이스에서 분류기 모델들을 생성할 수도 있다. 이러한 로컬 분류기 모델들은 디바이스 프로세서로 하여금 그 특정 디바이스 상에서의 거동이 비-양성인지 (예컨대, 그 디바이스의 성능에 있어서의 열화에 기여하는지) 의 여부를 결정할 때 가장 중요한 특정 피처들을 정확하게 식별할 수 있게 한다. 로컬 분류기 모델들은 또한 디바이스 프로세서로 하여금 그 특정 디바이스에서 거동을 분류하는 것에 대한 그 상대적인 중요도에 따라 테스트 또는 평가된 피처들을 우선순위화할 수 있게 한다.

[0089] 디바이스-특정형 분류기 모델은 특정 컴퓨팅 디바이스에서 활동 또는 거동을 분류하는 데 가장 관련되는 것으로 결정되는 컴퓨팅 디바이스-특정형 피처들/엔트리들만을 포함/테스트하는 포커싱된 데이터 모델을 포함하는 분류기 모델이다. 애플리케이션-특정형 분류기 모델은 특정 소프트웨어 애플리케이션을 평가하는 데 가장 관련되는 피처들/엔트리들만을 포함/테스트하는 포커싱된 데이터 모델을 포함하는 분류기 모델이다. 컴퓨팅 디바이스에서 국소적으로 애플리케이션-특정형 분류기 모델들을 동적으로 생성함으로써, 다양한 실시형태들은, 디바이스 프로세서로 하여금 특정 소프트웨어 애플리케이션의 동작들이 그 디바이스의 바람직하지 않거나 성능을 열화시키는 거동에 기여하고 있는지의 여부를 결정하기 위해 가장 중요한 소수의 피처들에 그 모니터링 및 분석 동작들을 포커싱할 수 있게 한다.

[0090] 다중-애플리케이션 분류기 모델은 둘 이상의 특정 소프트웨어 애플리케이션들 (또는 특정 유형들의 소프트웨어

애플리케이션들) 의 집합적 거동이 비-양성인지의 여부를 결정하는 데 가장 관련되는 피처들/엔트리들에 대한 테스트들을 포함하거나 우선순위화하는 포커싱된 데이터 모델을 포함하는 로컬 분류기 모델일 수도 있다. 다중-애플리케이션 분류기 모델은 집성된 피처 세트 및/또는 피처들의 집성된 세트를 테스트/평가하는 판단 노드들을 포함할 수도 있다. 디바이스 프로세서는 컴퓨팅 디바이스 상에서 동작하는 둘 이상의 소프트웨어 애플리케이션들 간의 관계들, 상호작용들, 및/또는 통신들을 식별하는 데 가장 관련되는 디바이스 피처들을 식별함으로써, 식별된 디바이스 피처들 중 하나를 평가하는 테스트 조건들을 식별함으로써, 식별된 테스트 조건들의 우선순위, 중요도, 또는 성공률들을 결정함으로써, 그들의 중요도 또는 성공률들에 따라 그 식별된 테스트 조건들을 우선순위화하거나 순서화함으로써, 그리고 식별된 테스트 조건들이 그 결정된 우선순위들, 중요도, 또는 성공률들에 따라 순서화되도록 식별된 테스트 조건들을 포함하는 분류기 모델을 생성함으로써, 다중-애플리케이션 분류기 모델을 생성하도록 구성될 수도 있다. 디바이스 프로세서는 또한 둘 이상의 애플리케이션-특정형 분류기 모델들을 결합함으로써 다중-애플리케이션 분류기 모델을 생성하도록 구성될 수도 있다.

[0091] 다양한 실시형태들에서, 디바이스 프로세서는 둘 이상의 애플리케이션들이 담합하거나 협력 작업한다고 또는 애플리케이션들이 일 그룹으로서 함께 분석되어야 한다고 결정하는 것에 응답하여 다중-애플리케이션 분류기 모델을 생성하도록 구성될 수도 있다. 디바이스 프로세서는 각각의 식별된 그룹 또는 애플리케이션들의 클래스에 대한 다중-애플리케이션 분류기 모델을 생성하도록 구성될 수도 있다. 그러나, 모든 그룹을 분석하면 상당한 양의 디바이스의 제한된 리소스들이 소모될 수도 있다. 따라서, 일 실시형태에서, 디바이스 프로세서는 일 애플리케이션이 (예컨대, 다른 애플리케이션들과의 상호작용들에 기초하여 등) 담합적 거동에 관여할 확률을 결정하고, 담합적 거동의 높은 확률이 존재하는 소프트웨어 애플리케이션들을 포함하는 그룹들만에 대한 분류기 모델을 지능적으로 생성하도록 구성될 수도 있다.

[0092] 거동 분석기 모듈 (208) 은 거동 추출기 모듈 (204) 에 의해 생성된 거동 벡터들을 분류기 모델에 적용하여, 모니터링된 활동 (또는 거동) 이 양성인지 또는 비-양성인지를 결정하도록 구성될 수도 있다. 일 실시형태에서, 거동 분석기 모듈 (208) 은, 그의 거동 분석 동작들의 결과들이 거동을 양성 또는 비-양성 중 어느 일방으로서 분류하기에 충분한 정보를 제공하지 않는 경우, 거동을 "의심스러운" 것으로서 분류할 수도 있다.

[0093] 거동 분석기 모듈 (208) 은 담합한 소프트웨어 애플리케이션들을 식별하고, 특정 애플리케이션들이 일 그룹으로서 평가되어야 한다고 결정하는 것에 응답하여, 그리고/또는 모니터된 활동 또는 거동이 의심스럽다고 결정하는 것에 응답하여, 거동 관측기 모듈 (202) 에 통지하도록 구성될 수도 있다. 이에 응답하여, 거동 관측기 모듈 (202) 은, 거동 분석기 모듈 (208) 로부터 수신된 정보 (예컨대, 실시간 분석 동작들의 결과들) 에 기초하여 그 관측들의 세분화도 (granularity) (즉, 컴퓨팅 디바이스 피처들이 모니터링되는 상세함의 레벨) 를 조정하고 그리고/또는 모니터링되는 애플리케이션들/요인들/거동들을 변경하고, 새로운 또는 부가적인 거동 정보를 생성 또는 수집하고, 추가적인 분석/분류를 위해 거동 분석기 모듈 (208) 에 새로운/부가적인 정보를 송신할 수도 있다.

[0094] 거동 관측기 모듈 (202) 및 거동 분석기 모듈 (208) 간의 이러한 피드백 통신들은 모바일 컴퓨팅 디바이스 (102) 로 하여금 관측들의 세분화도를 귀납적으로 증가시킬 수 있게 하는데 (즉, 보다 미세하거나 보다 상세한 관측들을 행할 수 있게 하고), 즉, 집합적 거동이 양성 또는 비-양성으로서 분류되어, 의심스럽거나 성능-열화된 거동의 원인이 식별될 때까지, 프로세싱 또는 배터리 소비 임계치에 도달할 때까지 또는 의심스럽거나 성능-열화된 거동이 관측 세분화도에 대한 추가적인 변경들, 조정들, 또는 증가들로부터 식별될 수 없다고 디바이스 프로세서가 결정할 때까지, 관측되는 피처들/거동들을 변경시킬 수 있게 한다. 이러한 피드백 통신은 또한 모바일 컴퓨팅 디바이스 (102) 가 과도한 양의 컴퓨팅 디바이스의 프로세싱, 메모리, 또는 에너지 리소스들을 소비하지 않으면서 거동 벡터들 및 분류기 모델들을 조정 또는 수정할 수 있게 한다.

[0095] 거동 관측기 모듈 (202) 및 거동 분석기 모듈 (208) 은 개별적으로 아니면 집합적으로 컴퓨팅 시스템의 거동들에 대한 실시간 거동 분석을 제공하여, 제한된 그리고 대략적 관측들로부터 의심스러운 거동을 식별하고, 관측할 거동들을 보다 상세히 동적으로 결정하며, 관측들에 대해 요구되는 상세 레벨을 동적으로 결정할 수도 있다.

이것은 모바일 컴퓨팅 디바이스 (102) 로 하여금 디바이스 상에서의 많은 양의 프로세서, 메모리, 또는 배터리 리소스들을 요구하지 않으면서, 문제들을 효율적으로 식별 및 방지할 수 있게 한다.

[0096] 다양한 실시형태들에서, 모바일 컴퓨팅 디바이스 (102) 의 디바이스 프로세서는, 긴밀한 모니터링을 요구하는 중요한 데이터 리소스를 식별하고, 중요한 데이터 리소스에 액세스 할 때 소프트웨어 애플리케이션들에 의해 만들어진 API 호출들을 (예컨대, 거동 관측기 모듈 (202) 을 통해) 모니터링하고, API 호출들의 패턴을 둘 이상의 소프트웨어 애플리케이션들에 의한 비-양성의 거동을 나타내는 것으로서 식별하고, API 호출들 및 리소스 사용

의 식별된 패턴에 기초하여 거동 벡터를 생성하고, 거동 벡터를 사용하여 거동 분석 동작들을 (예컨대, 거동 분석기 모듈 (208) 을 통해) 수행하고, 거동 분석 동작들에 기초하여 하나 이상의 소프트웨어 애플리케이션이 비-양성인지의 여부를 결정하도록 구성될 수도 있다.

[0097] 일 실시형태에서, 디바이스 프로세서는, 컴퓨팅 디바이스 상에서 동작하는 소프트웨어 애플리케이션들에 의해 가장 빈번하게 사용되는 API들을 식별하고, 식별된 핫 API들 (hot APIs) 의 사용에 관한 정보를 디바이스의 메모리 내의 API 로그에 저장하고, 비-양성의 거동을 식별하기 위해 API 로그에 저장된 정보에 기초하여 거동 분석 동작들을 수행하도록 구성될 수도 있다.

[0098] 다양한 실시형태들에서, 모바일 컴퓨팅 디바이스 (102) 는 활동 또는 거동이 비-양성인지의 여부를 결정하는 것과 가장 관련되는 피처들, 요인들, 및 데이터 포인트들을 지능적으로 및 효율적으로 식별하기 위해 네트워크 서버와 함께 작동하도록 구성될 수도 있다. 예를 들어, 디바이스 프로세서는 네트워크 서버로부터 전체 분류기 모델을 수신하고, 수신된 전체 분류기 모델을 사용하여 컴퓨팅 디바이스의 피처들 및 기능성들 또는 디바이스 상에서 동작하는 소프트웨어 애플리케이션들에 대해 특정된 희박 분류기 모델들 (즉, 데이터/거동 모델들) 을 생성하도록 구성될 수도 있다. 디바이스 프로세서는 전체 분류기 모델을 사용하여, 변동되는 레벨들의 복잡도 (또는 "희박도 (leanness)") 의 희박 분류기 모델들의 패밀리를 생성할 수도 있다. 가장 희박한 패밀리의 희박 분류기 모델들 (즉, 가장 적은 수의 테스트 조건들에 기초한 희박 분류기 모델) 은, 모델이 양성인 것으로도 양성이 아닌 것으로도 카테고리화할 수 없는 거동이 발생할 때까지 일률적으로 적용될 수도 있고 (따라서, 의심스러운 모델로 카테고리화되고), 그 때에는 거동을 카테고리화하기 위한 시도로 보다 로버스트한 (즉, 덜 희박한) 희박 분류기 모델이 적용될 수도 있다. 생성된 희박 분류기 모델들의 패밀리 내에서 훨씬 더 로버스트한 희박 분류기 모델들의 애플리케이션은 거동의 명확한 분류가 달성될 때까지 적용될 수도 있다.

이러한 방식으로, 디바이스 프로세서는 가장 완전하지만 리소스-집약적 희박 분류기 모델들의 사용을 로버스트한 분류기 모델이 동작을 명확하게 분류하는 데 필요하게 되는 그러한 상황들로 제한함으로써 효율성 및 정확성 사이에서의 균형을 유지할 수 있다.

[0099] 다양한 실시형태들에서, 디바이스 프로세서는 전체 분류기 모델에 포함된 유한 상태 머신 표시/표현을 부스트된 판단 스텀프들로 변환함으로써 희박 분류기 모델들을 생성하도록 구성될 수도 있다. 디바이스 프로세서는 전체 분류기 모델에 포함된 부스트된 판단 스텀프들의 서브세트를 포함하는 분류기 모델을 생성하기 위해 디바이스-특정형 피처들, 조건들, 또는 구성들에 기초하여 부스트된 판단 스텀프들의 전체 세트를 프루닝 (prune) 또는 컬링 (cull) 할 수도 있다. 그 다음, 디바이스 프로세서는 희박 분류기 모델을 사용하여 컴퓨팅 디바이스 거동을 지능적으로 모니터링, 분석 및/또는 분류할 수도 있다.

[0100] 부스트된 판단 스텀프들은 정확히 하나의 노드 (따라서 하나의 테스트 문항 또는 테스트 조건) 및 가중치를 갖는 하나의 레벨 판단 트리들 (trees) 이며, 따라서 데이터/거동들의 이진 분류 (binary classification) 에서 사용하기에 매우 적합하다. 즉, 증가된 판단 스텀프 결과들에 거동 벡터를 적용하면 이진 답변 (예컨대, 예 또는 아니오) 으로 귀착된다. 예를 들어, 부스트된 판단 스텀프에 의해 테스트된 질문/조건이 "분당 x 보다 적은 단문 메시지 서비스 (SMS) 송신들의 빈도" 인 경우, 부스트된 판단 스텀프에 "3" 의 값을 적용하면, ("3 회 미만" 의 SMS 송신들에 대해) "예" 답변 또는 ("3 회 이상" 의 SMS 송신들에 대해) "아니오" 답변 중 어느 일방으로 귀착될 것이다.

[0101] 부스트된 판단 스텀프들은 매우 간단하고 원초적이기 때문에 (따라서 상당한 프로세싱 리소스들을 요구하지 않기 때문에) 효율적이다. 부스트된 판단 스텀프들은 또한 매우 병렬화가능하며, 따라서 많은 스텀프들이 (예컨대, 컴퓨팅 디바이스 내의 다수의 코어들 또는 프로세서들에 의해) 병렬로/동시에 적용되거나 테스트될 수도 있다.

[0102] 일 실시형태에서, 디바이스 프로세서는 전체 분류기 모델에 포함된 분류기 기준들의 서브세트 및 컴퓨팅 디바이스 구성, 기능성, 및 접속된/포함된 하드웨어에 관련된 피처들에 대응하는 이러한 분류기 기준들만을 포함하는 희박 분류기 모델을 생성하도록 구성될 수도 있다. 디바이스 프로세서는 이 희박 분류 모델(들)을 사용하여 디바이스에 존재하거나 관련된 이러한 피처들 및 기능들만 모니터링할 수도 있다. 그 다음, 디바이스 프로세서는 희박 분류기 모델(들)을 주기적으로 수정 또는 재생성하여 컴퓨팅 디바이스의 현재 상태 및 구성에 기초하여 다양한 피처들 및 대응 분류기 기준들을 포함 또는 제거할 수도 있다.

[0103] 일례로서, 디바이스 프로세서는 거동 모델들의 전체 피처 세트와 연관된 판단 스텀프들을 포함하는 대형의 부스트된 판단 스텀프 분류기 모델을 수신하고, 컴퓨팅 디바이스의 현재 구성, 기능성, 작동 상태 및/또는 접속된/포함된 하드웨어와 관련되는 대형 분류기 모델(들)로부터 피처들만을 선택하고, 선택된 피처들에 대응하는 부스

트된 판단 스텀프들의 서브세트를 희박 분류기 모델에 포함시킴으로써 대형 분류기 모델들로부터 하나 이상의 희박 분류기 모델들을 도출하도록 구성될 수도 있다. 이 실시형태에서, 컴퓨팅 디바이스와 관련된 피처들에 대응하는 분류기 기준들은 그 선택된 피처들 중 적어도 하나를 테스트하는 대형 분류기 모델에 포함된 이러한 부스트된 판단 스텀프들일 수도 있다. 그 다음, 디바이스 프로세서는 컴퓨팅 디바이스의 현재 상태 및 구성에 기초하여 다양한 피처들을 포함 또는 제거하도록 부스트된 판단 스텀프 희박 분류기 모델(들)을 주기적으로 수정 또는 재생성하여, 희박 분류기 모델이 애플리케이션 특정형 또는 디바이스-특정형 피처 부스트된 판단 스텀프들을 계속 포함할 수도 있다.

[0104] 게다가, 디바이스 프로세서는 또한 특정 소프트웨어 애플리케이션들 (Google® 지갑 및 eTrade®) 및/또는 특정 유형의 소프트웨어 애플리케이션 (예컨대, 게임들, 네비게이션, 금융, 뉴스, 생산성 등) 과 관련되는 조건들 또는 피처들을 식별하는 애플리케이션-특정형 분류기 모델들을 동적으로 생성할 수도 있다. 이러한 분류기 모델들은 전체 분류기 모델에 포함되는 판단 노드들의 (또는 수신된 전체 분류기 모델로부터 생성된 보다 희박한 분류기 모델에 포함된 것들의), 감소되고 보다 포커싱된 서브세트를 포함하도록 생성될 수도 있다. 이러한 분류기 모델들은 다중-애플리케이션 분류기 모델들을 생성하도록 결합될 수도 있다.

[0105] 다양한 실시형태들에서, 디바이스 프로세서는 시스템에서의 각각의 소프트웨어 애플리케이션에 대해 그리고/또는 시스템에서의 소프트웨어 애플리케이션의 각각의 유형에 대해 애플리케이션-기반형 분류기 모델들을 생성하도록 구성될 수도 있다. 디바이스 프로세서는 또한 높은 위험성이 있거나 악용되기 쉬운 소프트웨어 애플리케이션들 및/또는 애플리케이션 유형들 (예컨대, 금융 애플리케이션들, POS (point-of-sale) 애플리케이션들, 생체인식 센서 애플리케이션들 등) 을 동적으로 식별하고, 높은 위험성이 있거나 악용되기 쉬운 것으로서 식별되는 소프트웨어 애플리케이션들 및/또는 애플리케이션 유형들에 대해서만 애플리케이션-기반형 분류기 모델을 생성하도록 구성될 수도 있다. 다양한 실시형태들에서, 디바이스 프로세서는, 동적으로, 반응적으로, 사전적으로, 그리고/또는 새로운 애플리케이션이 설치되거나 업데이트될 때마다, 애플리케이션-기반형 분류기 모델들을 생성하도록 구성될 수도 있다.

[0106] 각각의 소프트웨어 애플리케이션은 일반적으로 컴퓨팅 디바이스 상에서 다수의 작업들 또는 활동들을 수행한다. 특정 작업들/활동들이 컴퓨팅 디바이스에서 수행되는 특정 실행 상태는, 거동 또는 활동이 추가적인 또는 보다 철저한 조사, 모니터링 및/또는 분석에 가치가 있는지의 여부에 대한 강력한 표시자 (indicator) 일 수도 있다. 이와 같이, 다양한 실시형태들에서, 디바이스 프로세서는 그 거동 모니터링 및 분석 동작들을 포커싱하기 위해 특정 작업들/활동들이 수행되는 실제 실행 상태들을 식별하는 정보를 사용하고, 활동이 중요한 활동인지 및/또는 활동이 비-양성인지의 여부를 보다 양호하게 결정하도록 구성될 수도 있다.

[0107] 다양한 실시형태들에서, 디바이스 프로세서는, 소프트웨어 애플리케이션에 의해 수행되는 활동들/작업들을, 이들 활동들/작업들이 수행되었던 실행 상태들과 연관시키도록 구성될 수도 있다. 예를 들어, 디바이스 프로세서는 실행 상태가 관련되는 소프트웨어의 피처들, 활동들, 또는 동작들 (예컨대, 위치 액세스, SMS 판독 동작들, 센서 액세스 등) 을 리스트화한 서브-벡터 또는 데이터-구조에서 그 장치화된 컴포넌트들을 모니터링함으로써 수집된 거동 정보를 포함하는 거동 벡터를 생성하도록 구성될 수도 있다. 일 실시형태에서, 이 서브-벡터/데이터-구조는, 각각의 피처/활동/동작이 관측되었던 실행 상태를 식별하는 쉐도우 피처 값 서브-벡터/데이터-구조와 연관되어 저장될 수도 있다. 예로서, 디바이스 프로세서는 "위치_배경 (location_background)" 데이터 필드를 포함하는 거동 벡터를 생성할 수도 있으며, 그 데이터 필드의 값은 소프트웨어 애플리케이션이 배경 상태에서 동작하고 있었을 때 위치 정보에 액세스했던 수 또는 레이트를 식별한다. 이것은 디바이스 프로세서로 하여금 컴퓨팅 디바이스의 다른 관측된/모니터링된 활동들과 독립하여 그리고/또는 이와 병렬로 이 실행 상태 정보를 분석할 수 있게 한다. 이러한 방식으로 거동 벡터를 생성하는 것은 또한 시스템으로 하여금 시간에 따라 정보 (예컨대, 빈도 또는 레이트) 를 집성할 수 있게 한다.

[0108] 다양한 실시형태들에서, 디바이스 프로세서는 모니터링된 활동에 관한 질의에 대한 답변을 생성하기 위해 머신 학습 분류기 내의 판단 노드에 입력될 수도 있는 정보를 포함하도록 거동 벡터를 생성하도록 구성될 수도 있다.

[0109] 다양한 실시형태들에서, 디바이스 프로세서는 실행 정보를 포함하도록 거동 벡터들을 생성하도록 구성될 수도 있다. 실행 정보는 거동의 일부로서 (예컨대, 배경 프로세스에 의해 3 초에 5 번 사용된 카메라, 전경 프로세스에 의해 3 초에 3 번 사용된 카메라 등) 또는 독립적인 피처의 일부로서 거동 벡터에 포함될 수도 있다. 일 실시형태에서, 실행 상태 정보는 쉐도우 피처 값 서브-벡터 또는 데이터 구조로서 거동 벡터에 포함될 수도 있다. 일 실시형태에서, 거동 벡터는 실행 상태가 관련된 피처들, 활동들, 작업들과 연관되어 쉐도우 (shadow) 피처 값 서브-벡터/데이터 구조를 저장할 수도 있다.

[0110] 도 3 은 일 실시형태에 따라 둘 이상의 소프트웨어 애플리케이션들의 집합적 거동을 평가하기 위해 거동 분석 기법들을 사용하는 방법 (300) 을 나타낸다. 방법 (300) 은 모바일 또는 리소스 제약형 컴퓨팅 디바이스의 프로세싱 코어에서 수행될 수도 있다.

[0111] 블록 (302) 에서, 프로세싱 코어는 디바이스 상에서 동작하는 소프트웨어 애플리케이션들의 활동들을 모니터링할 수도 있다. 블록 (304) 에서, 프로세싱 코어는 모니터링된 활동들로부터 거동 정보를 수집할 수도 있다. 블록 (306) 에서, 프로세싱 코어는 수집된 거동 정보에 기초하여 거동 벡터를 생성할 수도 있다. 블록 (308) 에서, 프로세싱 코어는 분석 정보를 생성하기 위해 거동 벡터를 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용할 수도 있다. 블록 (310) 에서, 프로세싱 코어는 분석 정보를 사용하여 소프트웨어 애플리케이션들 간의 관계를 식별할 수도 있다. 블록 (312) 에서, 프로세싱 코어는 식별된 관계에 기초하여 일 그룹으로서 함께 평가되어야 하는 소프트웨어 애플리케이션들을 식별할 수도 있다. 블록 (314) 에서, 프로세싱 코어는 식별된 소프트웨어 애플리케이션들의 거동 정보 및/또는 분석 결과들을 집성할 수도 있다. 블록 (316) 에서, 프로세싱 코어는 집성된 분석 결과들을 사용하여 소프트웨어 애플리케이션들의 집합적 거동이 양성인지 또는 비-양성인지를 결정할 수도 있다.

[0112] 도 4 는 일 실시형태에 따른 소프트웨어 애플리케이션들 간의 관계를 결정하기 위해 거동 분석 기법들을 사용하는 방법 (400) 을 도시한다. 방법 (400) 은 모바일 또는 리소스 제약형 컴퓨팅 디바이스의 프로세싱 코어에서 수행될 수도 있다. 블록 (402) 에서, 프로세싱 코어는 컴퓨팅 디바이스 상에서 동작하는 소프트웨어 애플리케이션들 간의 상호작용들을 모니터링할 수도 있다. 블록 (404) 에서, 프로세싱 코어는 소프트웨어 애플리케이션들 간의 관계들을 특징화하는 거동 벡터들을 생성할 수도 있다. 블록 (406) 에서, 프로세싱 코어는 분석 정보를 생성하기 위해 거동 벡터들을 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용할 수도 있다. 블록 (408) 에서, 프로세싱 코어는 분석 정보를 사용하여 애플리케이션들 간의 관계의 성질, 이를테면 애플리케이션들이 담합 또는 협력 작업하고 있는지의 여부를 결정할 수도 있다.

[0113] 도 5 는, 일 실시형태에 따라 거동 분석 기법들을 사용하여, 그 식별된 애플리케이션들의 집합적 거동이 비-양심적인지의 여부를 결정하는 방법 (500) 을 나타낸다. 방법 (500) 은 모바일 또는 리소스 제약형 컴퓨팅 디바이스의 프로세싱 코어에서 수행될 수도 있다. 블록 (502) 에서, 프로세싱 코어는 일 그룹으로서 함께 분석되어야 하는 소프트웨어 애플리케이션들 (예컨대, 담합한 애플리케이션들 등) 을 식별할 수도 있다. 블록 (504) 에서, 프로세싱 코어는 식별된 애플리케이션의 거동 벡터들을 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용할 수도 있다. 블록 (506) 에서, 프로세싱 코어는 거동 벡터의 각 애플리케이션에 의해 생성된 분석 정보를 분류기 모델에 집성할 수도 있다. 블록 (508) 에서, 프로세싱 코어는 집성된 분석 정보를 사용하여 그 식별된 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정할 수도 있다.

[0114] 도 6 은 다른 실시형태에 따라 거동 분석 기법들을 사용하여 그 식별된 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정하는 방법 (600) 을 나타낸다. 방법 (600) 은 모바일 또는 리소스 제약형 컴퓨팅 디바이스의 프로세싱 코어에서 수행될 수도 있다. 블록 (602) 에서, 프로세싱 코어는 일 그룹으로서 함께 분석되어야 하는 소프트웨어 애플리케이션들을 식별할 수도 있다. 블록 (604) 에서, 프로세싱 코어는 식별된 애플리케이션들의 활동들을 모니터링할 수도 있다. 블록 (606) 에서, 프로세싱 코어는 모니터링된 활동들의 각각에 대한 거동 정보를 수집할 수도 있다. 블록 (608) 에서, 프로세싱 코어는 수집된 거동 정보에 기초하여 그 식별된 애플리케이션들의 집합적 거동을 특징화하는 거동 벡터를 생성할 수도 있다. 블록 (610) 에서, 프로세싱 코어는 생성된 거동 벡터를 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용하여 분석 정보를 생성할 수도 있다. 블록 (612) 에서, 프로세싱 코어는 분석 정보를 사용하여 그 식별된 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정할 수도 있다.

[0115] 도 7 은 컴퓨팅 디바이스의 거동을 분류하기 위해 희박 분류기 모델의 패밀리를 사용하는 일 실시형태 방법 (700) 을 나타낸다. 방법 (700) 은 모바일 또는 리소스 제약형 컴퓨팅 디바이스의 프로세싱 코어에 의해 수행될 수도 있다.

[0116] 블록 (702) 에서, 프로세싱 코어는 관측들을 수행하여 다양한 레벨들의 컴퓨팅 디바이스 시스템에서 장치화되는 다양한 컴포넌트들로부터 거동 정보를 수집할 수도 있다. 일 실시형태에서, 이것은 도 2 를 참조하여 위에서 논의된 거동 관측기 모듈 (202) 을 통해 달성될 수도 있다. 블록 (704) 에서, 프로세싱 코어는 관측들, 수집된 거동 정보, 및/또는 컴퓨팅 디바이스 거동을 특징화하는 거동 벡터를 생성할 수도 있다. 또한 블록 (704) 에서, 프로세싱 코어는 네트워크 서버로부터 수신된 전체 분류기 모델을 사용하여 다양한 레벨들의 복잡도 (또는 "희박도") 의 희박 분류기 모델 또는 희박 분류기 모델들의 일 패밀리를 생성할 수도 있다. 이를

달성하기 위해, 프로세싱 코어는 전체 분류기 모델에 포함된 부스트된 판단 스텀프들의 일 패밀리를 컬링하여, 감소된 수의 부스트된 판단 스텀프들을 포함하고/하거나 제한된 수의 테스트 조건들을 평가하는 희박 분류기 모델들을 생성할 수도 있다.

[0117] 블록 (706) 에서, 프로세싱 코어는 컴퓨팅 디바이스에 의해 아직 평가되지 않았거나 적용되지 않은 희박 분류기 모델들의 패밀리 (즉, 가장 적은 수의 상이한 컴퓨팅 디바이스 상태들, 피처들, 거동들, 또는 조건들에 기초한 모델) 에서 가장 희박한 분류기를 선택할 수도 있다. 일 실시형태에서, 이것은 분류 모델들의 순서화된 리스트에서 제 1 분류기 모델을 선택하는 프로세싱 코어에 의해 달성될 수도 있다.

[0118] 블록 (708) 에서, 프로세싱 코어는 수집된 거동 정보 또는 거동 벡터들을 그 선택된 희박 분류기 모델에서의 각각의 부스트된 판단 스텀프에 적용할 수도 있다. 부스트된 판단 스텀프들은 이진 결정들이고 희박 분류기 모델은 동일한 테스트 조건에 기초한 많은 이진 결정들을 선택함으로써 생성되기 때문에, 희박 분류기 모델에서 부스트된 판단 스텀프들에 거동 벡터를 적용하는 프로세스는 병렬 동작으로 수행될 수도 있다. 대안적으로, 블록 (530) 에서 적용된 거동 벡터는 희박 분류기 모델 내에 포함된 제한된 수의 테스트 조건 파라미터들을 단지 포함하기 위해 절단되거나 필터링될 수도 있으며, 이에 의해, 모델을 적용함에 있어서 연산 노력을 더 감소시킨다.

[0119] 블록 (710) 에서, 프로세싱 코어는 수집된 거동 정보를 희박 분류기 모델에서의 각각의 부스트된 판단 스텀프에 적용한 결과들의 가중 평균치를 연산하거나 결정할 수도 있다. 블록 (712) 에서, 프로세싱 코어는 연산된 가중 평균치를 임계 값과 비교할 수도 있다. 결정 블록 (714) 에서, 프로세싱 코어는 이 비교의 결과들 및/또는 선택된 희박 분류기 모델을 적용함으로써 생성된 결과들이 의심스러운지의 여부를 결정할 수도 있다. 예를 들어, 프로세싱 코어는 이들 결과들이 높은 신뢰성의 정도로 악성 또는 양성 중 어느 일방으로서 분류하는데 사용될 수도 있는지의 여부를 결정할 수도 있으며, 그렇지 않을 경우, 거동을 의심스러운 것으로서 취급할 수도 있다.

[0120] 결과들이 의심스럽다고 프로세싱 코어가 결정한다면 (예컨대, 결정 블록 (714) = "예"), 프로세싱 코어는 블록들 (706-712) 에서의 동작들을 반복하여, 높은 신뢰성의 정도로 악성 또는 양성으로서 분류될 때까지 더 많은 디바이스 상태들, 피처들, 거동들, 또는 조건들을 평가하는 더 강력한 (즉, 덜 희박한) 분류기 모델을 선택 및 적용할 수도 있다. 이를테면 거동이 높은 신뢰성의 정도로 악성 또는 양성 중 어느 일방으로서 분류될 수 있다고 결정함으로써, 결과들이 의심스럽지 않다고 프로세싱 코어가 결정한다면 (예컨대, 결정 블록 (714) = "아니오"), 블록 (716) 에서, 프로세싱 코어는 블록 (712) 에서 생성된 비교의 결과를 사용하여 컴퓨팅 디바이스의 거동을 양성 또는 잠재적으로 악성으로서 분류할 수도 있다.

[0121] 대안적인 실시형태 방법에서, 전술한 동작들은, 희박 분류기 모델에 아직 존재하지 않는 부스트된 판단 스텀프를 순차적으로 선택하고; 동일한 컴퓨팅 디바이스 상태, 피처, 거동, 또는 조건에 의존하는 (따라서, 하나의 결정 결과에 기초하여 적용될 수 있는) 다른 모든 부스트된 판단 스텀프들을 그 선택된 판단 스텀프로서 식별하고; 동일한 컴퓨팅 디바이스의 상태, 피처, 거동, 또는 조건에 의존하는 그 선택된 그리고 모든 식별된 다른 부스트된 판단 스텀프들을 희박 분류기 모델에 포함시키고; 그 결정된 수의 테스트 조건들과 동일한 횟수로 프로세스를 반복함으로써 달성될 수도 있다. 선택된 부스트된 판단 스텀프와 동일한 테스트 조건에 의존하는 모든 부스트된 판단 스텀프들이 그 때마다 희박 분류기 모델에 부가되기 때문에, 이 프로세스가 수행되는 횟수를 제한하는 것은 희박 분류기 모델에 포함된 테스트 조건들의 수를 제한할 것이다.

[0122] 도 8 은 다양한 실시형태들에 따라 사용에 적합한 부스트된 판단 트리/분류기를 생성하기에 적합한 예시적인 부스팅 방법 (800) 을 나타낸다. 블록 (802) 에서, 프로세서는 판단 트리/분류기를 생성 및/또는 실행하고, 판단 트리/분류기의 실행으로부터 트레이닝 샘플을 수집하고, 트레이닝 샘플에 기초하여 새로운 분류기 모델 (h1(x)) 을 생성할 수도 있다. 트레이닝 샘플은 컴퓨팅 디바이스의 컴퓨팅 디바이스 거동들, 소프트웨어 애플리케이션들, 또는 프로세스들에 대한 이전의 관측들 또는 분석으로부터 수집된 정보를 포함할 수도 있다. 트레이닝 샘플 및/또는 새로운 분류기 모델 (h1(x)) 은 이전의 분류기들에 포함된 질문 또는 테스트 조건들의 유형들에 기초하여, 그리고/또는 거동 분석기 모듈 (208) 의 분류기 모듈에서의 분류기들 또는 이전의 데이터/거동 모델들의 실행/적용으로부터 수집된 정확도 또는 성능 특성들에 기초하여 생성될 수도 있다. 블록 (804) 에서, 프로세서는 그 생성된 판단 트리/분류기 (h1(x)) 에 의해 오분류 (misclassify) 되었던 엔트리들의 가중치를 부스트 (또는 증가) 시켜서, 제 2 의 새로운 트리/분류기 (h2(x)) 를 생성할 수도 있다. 일 실시형태에서, 트레이닝 샘플 및/또는 새로운 분류기 모델 (h2(x)) 은 분류기의 이전의 실행 또는 사용 (h1(x)) 의 오류 레이트에 기초하여 생성될 수도 있다. 일 실시형태에서, 트레이닝 샘플 및/또는 새로운 분류기 모델

(h2(x)) 은 분류기의 이전의 실행 또는 사용에서 데이터 포인트들의 오분류 또는 오류 레이트에 기여된 것을 갖는 것으로 결정된 속성들에 기초하여 생성될 수도 있다.

[0123] 일 실시형태에서, 오분류된 엔트리들은 그들의 상대적 정확도 또는 유효성에 기초하여 가중화될 수도 있다. 블록 (806) 에서, 프로세서는 그 생성된 제 2 트리/분류기 (h2(x)) 에 의해 오분류되었던 엔트리들의 가중치를 부스트 (또는 증가) 시켜서, 제 3 의 새로운 트리/분류기 (h3(x)) 를 생성할 수도 있다. 블록 (808) 에서, 블록들 (804-806) 의 동작들은 "t" 개의 새로운 트리/분류기들 ($h_t(x)$) 을 생성하기 위해 반복될 수도 있다.

[0124] 제 2 트리/분류기 (h2(x)) 는, 제 1 판단 트리/분류기 (h1(x)) 에 의해 오분류되었던 엔트리들의 가중치를 부스트하거나 증가시킴으로써, 제 1 판단 트리/분류기 (h1(x)) 에 의해 오분류되었던 엔티티들을 보다 정확히 분류할 수도 있지만, 또한 제 1 판단 트리/분류기 (h1(x)) 에 의해 올바르게 분류되었던 엔티티들 중 일부를 오분류할 수도 있다. 유사하게, 제 3 트리/분류기 (h3(x)) 는 제 2 판단 트리/분류기 (h2(x)) 에 의해 오분류되었던 엔티티들을 보다 정확하게 분류하고, 제 2 판단 트리/분류기 (h2(x)) 에 의해 올바르게 분류되었던 엔티티들 중 일부를 오분류할 수도 있다. 즉, 트리/분류기들 $h1(x) - h_t(x)$ 의 패밀리를 생성하는 것은 전체적으로 수렴하는 시스템으로 귀착되지 않을 수도 있지만, 병렬로 실행될 수도 있는 다수의 판단 트리들/분류기들로 귀착된다.

[0125] 도 9 는 일 실시형태에 따라 동적 및 적응 관측들을 수행하도록 구성된 컴퓨팅 시스템의 거동 관측기 모듈 (202) 에서의 예시적인 논리 컴포넌트들 및 정보 흐름들을 나타낸다. 거동 관측기 모듈 (202) 은 적응 필터 모듈 (902), 스로틀 (throttle) 모듈 (904), 관측기 모드 모듈 (906), 하이-레벨 거동 검출 모듈 (908), 거동 벡터 생성기 (910), 및 보안 버퍼 (912) 를 포함할 수도 있다. 하이-레벨 거동 검출 모듈 (908) 은 공간적 상관 (correlation) 모듈 (914) 및 시간적 상관 모듈 (916) 을 포함할 수도 있다.

[0126] 관측기 모드 모듈 (906) 은 분석기 유닛 (예컨대, 도 2 를 참조하여 전술한 거동 분석기 모듈 (208)) 및/또는 애플리케이션 API 를 포함할 수도 있는 다양한 소스들로부터 제어 정보를 수신할 수도 있다. 관측기 모드 모듈 (906) 은 다양한 관측기 모드들에 관한 제어 정보를 적응 필터 모듈 (902) 및 하이-레벨 거동 검출 모듈 (908) 에 송신할 수도 있다.

[0127] 적응 필터 모듈 (902) 은 다수의 소스들로부터 데이터/정보를 수신하고, 수신된 정보를 지능적으로 필터링하여, 수신된 정보로부터 선택된 정보의 더 작은 서브세트를 생성할 수도 있다. 이 필터는 분석기 모듈 또는 API 를 통해 통신하는 하이-레벨 프로세스로부터 수신된 정보 또는 제어에 기초하여 적응될 수도 있다. 필터링된 정보는 스로틀 모듈 (904) 에 송신될 수도 있으며, 스로틀 모듈 (904) 은 필터로부터 흐르는 정보의 양을 제어하는 것을 담당하여, 하이-레벨 거동 검출 모듈 (908) 이 요청 또는 정보로 플러딩 (flood) 또는 오버로딩 (overload) 되지 않는 것을 보장할 수도 있다.

[0128] 하이-레벨 거동 검출 모듈 (908) 은 스로틀 모듈 (904) 로부터의 데이터/정보, 관측기 모드 모듈 (906) 로부터의 제어 정보, 및 컴퓨팅 디바이스의 다른 컴포넌트들로부터의 컨텍스트 정보를 수신할 수도 있다. 하이-레벨 거동 검출 모듈 (908) 은 수신된 정보를 사용하여 공간적 및 시간적 상관들을 수행하여, 디바이스로 하여금 차선의 (sub-optimal) 레벨들에서 수행하게 할 수도 있는 하이-레벨 거동들을 검출 및 식별할 수도 있다. 공간적 및 시간적 상관들의 결과들은 거동 벡터 생성기 (910) 에 송신될 수도 있고 이 거동 벡터 생성기 (910) 는 상관 정보를 수신할 수도 있고, 특정 프로세스, 애플리케이션, 또는 서브-시스템의 거동들을 설명하는 거동 벡터를 생성할 수 있다. 일 실시형태에서, 거동 벡터 생성기 (910) 는 특정 프로세스, 애플리케이션, 또는 서브-시스템의 각각의 하이-레벨 거동이 거동 벡터의 엘리먼트가 되도록 하는 거동 벡터를 생성할 수도 있다. 일 실시형태에서, 생성된 거동 벡터는 보안 버퍼 (912) 에 저장될 수도 있다. 하이-레벨 거동 검출의 예들로는 특정 이벤트의 존재, 다른 이벤트의 양 또는 빈도, 다수의 이벤트들 간의 관계, 이벤트들이 발생하는 순서, 어떤 이벤트들의 발생 사이의 시간 차이들 등을 포함할 수도 있다.

[0129] 다양한 실시형태들에서, 거동 관측기 모듈 (202) 은 적응적 관측들을 수행할 수도 있고 관측 세분화도를 제어할 수도 있다. 즉, 거동 관측기 모듈 (202) 은, 관측되어야 하는 관련 거동들을 동적으로 식별할 수도 있고, 식별된 거동들이 관측되어야 하는 상세함의 레벨을 동적으로 결정할 수도 있다. 이러한 방식으로, 거동 관측기 모듈 (202) 은, 시스템으로 하여금 다양한 레벨들 (예컨대, 다수의 대략적 그리고 미세한 레벨들) 에서 컴퓨팅 디바이스의 거동들을 모니터링할 수 있게 한다. 거동 관측기 모듈 (202) 은 시스템으로 하여금 관측되고 있는 것에 적응할 수 있게 할 수도 있다. 거동 관측기 모듈 (202) 은 시스템으로 하여금 광범위한 소스

들로부터 획득될 수도 있는 정보의 포커싱된 서브세트에 기초하여 관측되고 있는 요인들/거동들을 동적으로 변경시킬 수 있게 할 수도 있다.

[0130] 전술한 바와 같이, 거동 관측기 모듈 (202) 은 다양한 소스들로부터 수신된 정보에 기초하여 적응적 관측 기법들을 수행하고 관측 세분화도를 제어할 수도 있다. 예를 들어, 하이-레벨 거동 검출 모듈 (908) 은 스로틀 모듈 (904), 관측기 모드 모듈 (906), 및 컴퓨팅 디바이스의 다른 컴포넌트들 (예컨대, 센서들) 로부터 수신된 컨텍스트 정보로부터 정보를 수신할 수도 있다. 일례로서, 시간적 상관들을 수행하는 하이-레벨 거동 검출 모듈 (908) 은, 카메라가 사용되었고 컴퓨팅 디바이스가 사진 (picture) 을 서버에 업로드하려고 시도하고 있음을 검출할 수도 있다. 하이-레벨 거동 검출 모듈 (908) 은 또한 공간적 상관들을 수행하여, 그 디바이스가 수납되었고 사용자의 벨트에 수납되었거나 부착되었던 동안에 컴퓨팅 디바이스 상의 애플리케이션이 사진을 촬영했는지의 여부를 결정할 수도 있다. 하이-레벨 거동 검출 모듈 (908) 은 이러한 검출된 하이-레벨 거동 (예컨대, 수납된 동안의 카메라의 사용) 이, 컴퓨팅 디바이스의 현재의 거동을 과거의 거동과 비교함으로써 그리고/또는 복수의 디바이스들로부터 수집된 정보 (예컨대, 크라우드-소싱 서버로부터 수신된 정보) 에 액세스함으로써 달성될 수도 있는, 수용가능하거나 보편적인 거동인지의 여부를 결정할 수 있다. 수납된 동안에 사진들을 촬영하여 이를 서버에 업로드하는 것은 (수납되어 있다는 컨텍스트에서의 관측된 정상적인 거동들로부터 결정될 수도 있는 바와 같이) 드문 거동이므로, 이 상황에서 하이-레벨 거동 검출 모듈 (908) 은 이를 잠재적으로 위협적인 거동인 것으로서 인식하고 적절한 응답 (예컨대, 카메라 끄기, 경고음 울리기 등) 을 개시할 수도 있다.

[0131] 일 실시형태에서, 거동 관측기 모듈 (202) 은 다수의 부분들로 구현될 수도 있다.

[0132] 도 10 은 실시형태의 관측기 데몬을 구현하는 컴퓨팅 시스템 (1000) 에서의 논리적 컴포넌트들 및 정보 흐름들을 보다 상세히 나타낸다. 도 10 에 나타낸 예에서, 컴퓨팅 시스템 (1000) 은 거동 검출기 (1002) 모듈, 데이터베이스 엔진 (1004) 모듈, 및 거동 분석기 모듈 (208) 을 사용자 공간에 포함하고, 링 버퍼 (1014), 필터 규칙들 (1016) 모듈, 스로틀링 규칙들 (1018) 모듈, 및 보안 버퍼 (1020) 를 커널 공간에 포함한다. 컴퓨팅 시스템 (1000) 은, 사용자 공간에 거동 검출기 (1002) 및 데이터베이스 엔진 (1004) 을, 그리고 커널 공간에 보안 버퍼 관리기 (1006), 규칙들 관리기 (1008), 및 시스템 헬스 (health) 모니터 (1010) 를 포함하는 관측기 데몬을 더 포함할 수도 있다.

[0133] 다양한 실시형태들은 시스템 거동을 특징화하기 위해 웹킷 (webkit), SDK, NDK, 커널, 드라이버들, 및 하드웨어를 망라하는 컴퓨팅 디바이스에 대한 교차-계층 관측들을 제공할 수도 있다. 거동 관측들은 실시간으로 이루어질 수도 있다.

[0134] 관측기 모듈은 적응적 관측 기법들을 수행할 수도 있고 관측 세분화도를 제어할 수도 있다. 전술한 바와 같이, 컴퓨팅 디바이스의 열화에 기여할 수도 있는 상당히 많은 수의 (즉, 수천 개의) 요인들이 존재하며, 디바이스의 성능의 열화에 기여할 수도 있는 상이한 요인들의 전부를 모니터링/관측하는 것은 실현가능하지 않을 수도 있다. 이를 극복하기 위해, 다양한 실시형태들은 관측되어야 하는 관련 거동들을 동적으로 식별하고, 식별된 거동들이 관측되어야 하는 상세함의 레벨을 동적으로 결정한다.

[0135] 도 11 은 일 실시형태에 따라 동적 및 적응적 관측들을 수행하기 위한 예시적인 방법 (1100) 을 나타낸다. 블록 (1102) 에서, 디바이스 프로세서는 컴퓨팅 디바이스의 열화에 기여할 수도 있는 상당히 많은 수의 요인들/거동들의 일 서브세트를 모니터링/관측함으로써 대략적 관측들을 수행할 수도 있다. 블록 (1103) 에서, 디바이스 프로세서는 대략적 관측들에 기초하여 대략적 관측들 및/또는 컴퓨팅 디바이스 거동을 특징화하는 거동 벡터를 생성할 수도 있다. 블록 (1104) 에서, 디바이스 프로세서는 컴퓨팅 디바이스의 열화에 잠재적으로 기여할 수도 있는 대략적 관측들과 연관된 서브시스템들, 프로세스들, 및/또는 애플리케이션들을 식별할 수도 있다. 이것은, 예를 들어, 다수의 소스들로부터 수신된 정보를, 컴퓨팅 디바이스의 센서들로부터 수신된 컨텍스트 정보와 비교함으로써 달성될 수도 있다. 블록 (1106) 에서, 디바이스 프로세서는 대략적 관측들에 기초하여 거동 분석 동작들을 수행할 수도 있다. 일 실시형태에서, 블록들 (1103 및 1104) 의 일부로서, 디바이스 프로세서는 도 2-도 10 을 참조하여 위에서 논의된 동작들 중 하나 이상을 수행할 수도 있다.

[0136] 결정 블록 (1108) 에서, 디바이스 프로세서는 의심스러운 거동들 또는 잠재적인 문제들이 거동 분석의 결과들에 기초하여 식별되고 정정될 수 있는지의 여부를 결정할 수도 있다. 의심스러운 거동들 또는 잠재적인 문제들이 거동 분석의 결과들에 기초하여 식별 및 정정될 수 있다고 디바이스 프로세서가 결정하는 경우 (즉, 결정 블록 (1108) = "예"), 블록 (1118) 에서 프로세서는 거동을 정정하기 위한 프로세스를 개시하고, 부가적인 대략적 관측들을 수행하기 위해 블록 (1102) 으로 복귀할 수도 있다.

[0137] 디바이스 프로세서는 의심스러운 거동들 또는 잠재적인 문제들이 거동 분석의 결과들에 기초하여 식별 및/또는 정정될 수 없다고 결정하는 경우 (즉, 결정 블록 (1108) = "아니오"), 결정 블록 (1109) 에서 디바이스 프로세서는 문제의 가능성이 있는지의 여부를 결정할 수도 있다. 일 실시형태에서, 디바이스 프로세서는 컴퓨팅 디바이스가 잠재적인 문제들을 만나고 그리고/또는 의심스러운 거동들에 관여할 확률을 연산하고, 연산된 확률이 미리 결정된 임계치보다 큰지의 여부를 결정함으로써 문제의 가능성이 있음을 결정할 수도 있다. 연산된 확률이 미리 결정된 임계치보다 크지 않고 그리고/또는 의심스러운 거동들 또는 잠재적인 문제들이 존재하고 그리고/또는 검출가능할 가능성이 없다고 디바이스 프로세서가 결정하는 경우 (즉, 결정 블록 (1109) = "아니오"), 프로세서는 부가적인 대략적 관측들을 수행하기 위해 블록 (1102) 으로 복귀할 수도 있다.

[0138] 디바이스 프로세서는 의심스러운 거동들 또는 잠재적인 문제들이 존재하고 그리고/또는 검출가능할 가능성이 있다고 결정하는 경우 (즉, 결정 블록 (1109) = "예"), 블록 (1110) 에서 디바이스 프로세서는 식별된 서브시스템들, 프로세스들 또는 애플리케이션들에 대한 보다 심도 있는 로깅 (logging) /관측들 또는 최종적인 로깅을 수행할 수도 있다. 블록 (1112) 에서, 디바이스 프로세서는 식별된 서브시스템들, 프로세스들 또는 애플리케이션들에 대해 보다 심도 있고 보다 상세한 관측들을 수행할 수도 있다. 블록 (1114) 에서, 디바이스 프로세서는 보다 심도 있고 보다 상세한 관측들에 기초하여 추가적인 그리고/또는 보다 심도 있는 거동 분석을 수행할 수도 있다. 결정 블록 (1108) 에서, 디바이스 프로세서는 의심스러운 거동들 또는 잠재적 문제들이 보다 심도 있는 거동 분석의 결과들에 기초하여 식별되고 정정될 수 있는지의 여부를 다시 결정할 수도 있다. 의심스러운 거동들 또는 잠재적인 문제들이 보다 심도 있는 거동 분석의 결과들에 기초하여 식별 및 정정될 수 없다고 디바이스 프로세서가 결정하는 경우 (즉, 결정 블록 (1108) = "아니오"), 프로세서는, 상세함의 레벨이 문제를 식별하기에 충분히 미세할 때까지 또는 문제가 부가적인 상세함으로 식별될 수 없거나 아무런 문제가 존재하지 않는다고 결정될 때까지, 블록들 (1110-1114) 에서의 동작들을 반복할 수도 있다.

[0139] 의심스러운 거동들 또는 잠재적인 문제들이 보다 심도 있는 거동 분석의 결과들에 기초하여 식별 및 정정될 수 있다고 디바이스 프로세서가 결정하는 경우 (즉, 결정 블록 (1108) = "예"), 블록 (1118) 에서, 디바이스 프로세서는 문제/거동을 정정하기 위한 동작들을 수행할 수도 있고, 프로세서는 부가적인 동작들을 수행하기 위해 블록 (1102) 으로 복귀할 수도 있다.

[0140] 일 실시형태에서, 방법 (1100) 의 블록들 (1102-1118) 의 일부로서, 디바이스 프로세서는 제한된 그리고 대략적 관측들로부터 의심스러운 거동들을 식별하기 위해 시스템의 거동들의 실시간 거동 분석을 수행하여, 관측할 거동들을 보다 상세히 동적으로 결정하고, 관측들에 대해 요구되는 정확한 상세함의 레벨을 동적으로 결정할 수도 있다. 이것은 디바이스 프로세서로 하여금 디바이스 상에서 다량의 프로세서, 메모리, 또는 배터리 리소스들의 사용을 요구하지 않으면서, 문제들을 효율적으로 식별 및 방지할 수 있게 한다.

[0141] 다양한 실시형태들은 소프트웨어 애플리케이션들의 선택된 그룹의 집합적 거동을 모니터링 및 분석하기 위해 (권한들, 정책, 또는 규칙 기반의 접근법들과는 대조적으로) 거동 분석 및/또는 머신 학습 기법들을 사용함으로써 기존의 솔루션들을 개선한다. 현대의 컴퓨팅 디바이스들은 고도로 구성가능하고 복잡한 시스템들이기 때문에 거동 분석 또는 머신 학습 기법들을 사용하는 것이 중요하며, 소프트웨어 애플리케이션들이 담합하고 있는지의 여부를 결정하는 데 가장 중요한 요인들은 각 디바이스마다 상이할 수도 있다. 또한, 디바이스 피처들/요인들의 상이한 조합들은, 소프트웨어 애플리케이션들이 담합하고 있는지의 여부를 그 디바이스가 결정할 수 있도록 각 디바이스에서 일 분석을 요구할 수도 있다. 그러나, 모니터링 및 분석을 요구하는 피처들/요인들의 정확한 조합은, 오직, 활동이 수행되는 특정 컴퓨팅 디바이스로부터 획득된 정보를 사용하여 그리고 활동이 진행 중일 때 결정될 수 있다. 이러한 이유 및 다른 이유로 인해, 기존의 솔루션들은, 거동이 진행되는 동안 그리고 상당한 양의 컴퓨팅 디바이스의 프로세싱, 메모리, 또는 전력 리소스들을 소비하지 않은 채로, 컴퓨팅 디바이스 내의 복수의 소프트웨어 애플리케이션들의 집합적 거동 또는 이들 간의 관계들을 실시간으로 모니터링, 검출, 및 특징화하기에 적합하지 않다.

[0142] 다양한 실시형태들 (도 1-도 11 을 참조하여 전술한 실시형태들을 포함하지만 이에 한정되는 것은 아님) 은 다양한 컴퓨팅 디바이스들상 에서 구현될 수 있으며, 그 일례는 도 12 에서 스마트폰의 형태로 도시된다. 스마트폰 (1200) 은 내부 메모리 (1204), 디스플레이 (1212), 및 스피커 (1214) 에 커플링된 프로세서 (1202) 를 포함할 수도 있다. 또한, 스마트폰 (1200) 은 프로세서 (1202) 에 커플링된 무선 데이터 링크 및/또는 셀룰러 전화 트랜시버 (1208) 에 접속될 수도 있는 전자기 복사 (electromagnetic radiation) 를 송신 및 수신하기 위한 안테나를 포함할 수도 있다. 스마트폰 (1200) 은 또한 통상적으로 사용자 입력을 수신하기 위한 메뉴 선택 버튼들 또는 로커 (rocker) 스위치들 (1220) 을 포함한다.

[0143] 통상적인 스마트폰 (1200) 은 또한 마이크로폰으로부터 수신된 사운드를 무선 송신에 적합한 데이터 패킷들로 디지털화하고, 수신된 사운드 데이터 패킷들을 디코딩하여, 사운드를 생성하기 위해 스피커에 제공되는 아날로그 신호들을 생성하는 사운드 인코딩/디코딩 (CODEC) 회로 (1206) 를 포함한다. 또한, 프로세서 (1202), 무선 트랜시버 (1208) 및 CODEC (1206) 중 하나 이상은 디지털 신호 프로세서 (DSP) 회로 (별도로 도시되지 않음) 를 포함할 수도 있다. 일 실시형태에서, 프로세서 (1202) 는 도 1 에 도시된 SOC (100) 와 같은 시스템-온-칩 (SOC) 에 포함될 수도 있다. 일 실시형태에서, 프로세서 (1202) 는 도 1 에 도시된 애플리케이션 프로세서 (108) 일 수도 있다. 일 실시형태에서, 프로세서 (1202) 는 프로세싱 코어 (예컨대, IP 코어, CPU 코어 등) 일 수도 있다.

[0144] 실시형태 방법들의 일부분들은, 실시형태 방법들을 실행하는 동안 디바이스 프로세서에 의해 액세스될 수도 있는, 정상 동작의 거동들의 데이터베이스들을 유지하는 것과 같은, 서버에서 발생하는 프로세싱의 일부를 갖는 클라이언트-서버 아키텍처에서 달성될 수도 있다. 이러한 실시형태들은 도 13 에 도시된 서버 (1300) 와 같은 상업적으로 이용가능한 다양한 서버 디바이스들 중 어느 하나 상에서 구현될 수도 있다. 이러한 서버 (1300) 는 통상적으로 휘발성 메모리 (1302), 및 디스크 드라이브 (1303) 와 같은 대용량의 비휘발성 메모리에 커플링된 프로세서 (1301) 를 포함한다. 서버 (1300) 은 또한, 프로세서 (1301) 에 연결된 플로피 디스크 드라이브, 컴팩트 디스크 (CD) 또는 DVD 디스크 드라이브 (1304) 를 포함할 수도 있다. 서버 (1300) 는, 또한 다른 브로드캐스트 시스템 컴퓨터들 및 서버들에 커플링되는 로컬 영역 네트워크와 같은 네트워크 (1305) 와의 데이터 접속들을 확립하기 위해 프로세서 (1301) 에 커플링되는 네트워크 액세스 포트들 (1306) 을 포함할 수도 있다.

[0145] 프로세서 (1202, 1301) 는 후술하는 다양한 실시형태들의 기능들을 포함하는, 다양한 기능들을 수행하기 위해 소프트웨어 명령들 (애플리케이션들) 에 의해 구성될 수 있는 임의의 프로그램가능형 마이크로프로세서, 마이크로컴퓨터 또는 다중 프로세서 칩 또는 칩들일 수도 있다. 일부 모바일 디바이스들에서, 무선 통신 기능들에 전용되는 하나의 프로세서 및 다른 애플리케이션들을 구동하는 데 전용되는 하나의 프로세서와 같은, 다중 프로세서들 (1202) 이 제공될 수도 있다. 통상적으로, 소프트웨어 애플리케이션들은 액세스되고 프로세서 (1202, 1301) 에 로딩되기 전에 내부 메모리 (1204, 1302, 1303) 에 저장될 수도 있다. 프로세서 (1202, 1301) 는 애플리케이션 소프트웨어 명령들을 저장하기에 충분한 내부 메모리를 포함할 수도 있다.

[0146] 본 명세서에서 사용된 바와 같이, "컴포넌트", "모듈" 등과 같은 용어는, 특정 동작들 또는 기능들을 수행하도록 구성되는, 하드웨어, 펌웨어, 하드웨어와 소프트웨어의 조합, 소프트웨어, 또는 실행중인 소프트웨어와 같은, 하지만 이에 한정되는 것은 아닌, 컴퓨터 관련 엔티티를 포함하도록 의도된다. 예를 들어, 컴포넌트는 프로세서 상에서 구동되는 프로세스, 프로세서, 객체, 실행가능 파일, 실행의 스레드, 프로그램, 및/또는 컴퓨터일 수도 있지만, 이에 한정되는 것은 아니다. 예시로서, 컴퓨팅 디바이스에서 구동되는 애플리케이션 및 컴퓨팅 디바이스 양방 모두는 컴포넌트로서 지칭될 수도 있다. 하나 이상의 컴포넌트들이 프로세스 및/또는 실행의 스레드 내에 상주할 수도 있으며, 컴포넌트는 하나의 프로세서 또는 코어에 국한될 수도 있고 그리고/또는 둘 이상의 프로세서들 또는 코어들 사이에 분산될 수도 있다. 또한, 이들 컴포넌트들은 다양한 명령들 및/또는 데이터 구조들이 저장된 다양한 비-일시적 컴퓨터 판독가능 매체로부터 실행될 수도 있다. 컴포넌트들은 로컬 및/또는 원격 프로세스들, 기능 또는 프로시저 호출들, 전자 신호들, 데이터 패킷들, 메모리 판독/기록들, 및 다른 알려진 네트워크, 컴퓨터, 프로세서, 및/또는 프로세스 관련 통신 방법론들을 통해 통신할 수도 있다.

[0147] 다양한 실시형태들의 동작들을 수행하기 위한 프로그램가능형 프로세서 상의 실행을 위한 컴퓨터 프로그램 코드 또는 "프로그램 코드" 는, C, C++, C#, 스몰토크 (Smalltalk), 자바 (Java), 자바스크립트 (JavaScript), 비주얼 베이직 (Visual Basic), 구조화된 쿼리 언어 (예컨대, Transact-SQL), 펄 (Perl), 또는 다양한 다른 프로그래밍 언어들과 같은 하이 레벨 프로그래밍 언어로 기록될 수도 있다. 본 출원에서 사용되는 컴퓨터 판독가능 저장 매체 상에 저장된 프로그램 코드 또는 프로그램들은 그 포맷이 프로세서에 의해 이해가능한 머신 언어 코드 (이를테면, 오브젝트 코드) 를 지칭할 수도 있다.

[0148] 많은 모바일 컴퓨팅 디바이스 운영 체제 커널들은 (권한없는 코드가 구동되는) 사용자 공간 및 (권한있는 코드가 구동되는) 커널 공간으로 구성된다. 이러한 분리는 안드로이드 (Android®) 및 기타 일반 공용 라이센스 (GPL: general public license) 환경들에서 특히 중요하며, 이 환경들에서 커널 공간의 일부인 코드는 GPL 라이센스를 취득해야 하지만, 사용자 공간에서 구동되는 코드는 GPL 라이센스를 취득하지 않을 수도 있다. 여기에서 논의된 다양한 소프트웨어 컴포넌트들/모듈들은, 달리 명시적으로 언급되지 않는 한, 커널 공간 또는 사용

자 공간 중 어느 일방에서 구현될 수도 있음을 이해해야 한다.

[0149] 전술한 방법 설명들 및 프로세스 흐름도는 단지 예시적인 예들로서 제공되며, 다양한 실시형태들의 단계들이 제시된 순서로 수행되어야 하는 것을 요구하거나 암시하려고 의도되는 것은 아니다. 당업자에 의해 이해될 바와 같이, 전술한 실시형태들에서의 단계들의 순서는 임의의 순서로 수행될 수도 있다. "이후", "다음으로", "다음" 등과 같은 단어들은 단계들의 순서를 제한하도록 의도되지 않으며, 이들 단어들은 방법들의 설명을 통해 독자를 안내하는 데 단순히 사용된다. 또한, 예를 들어, 관사들 "a", "an", 또는 "the" 를 이용하는 단수인 청구항 엘리먼트들에 대한 임의의 참조는 엘리먼트를 단수로 제한하는 것으로 해석되어서는 아니된다.

[0150] 본원에 개시된 실시형태들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 회로들, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양방 모두의 조합으로서 구현될 수도 있다. 하드웨어와 소프트웨어의 이러한 호환성을 명확히 예시하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들, 및 단계들이 일반적으로 그들의 기능성의 측면에서 위에서 설명되었다. 이러한 기능성이 하드웨어로 구현될지 또는 소프트웨어로 구현될 지는 전체 시스템에 부과된 설계 제약들 및 특정 애플리케이션에 의존한다. 당업자는 각각의 특정 애플리케이션에 대해 다양한 방식들로 위에서 설명된 기능성을 구현할 수도 있지만, 이러한 구현형태의 결정들이 본 발명의 범위로부터의 이탈을 야기하는 것으로 해석되어서는 아니된다.

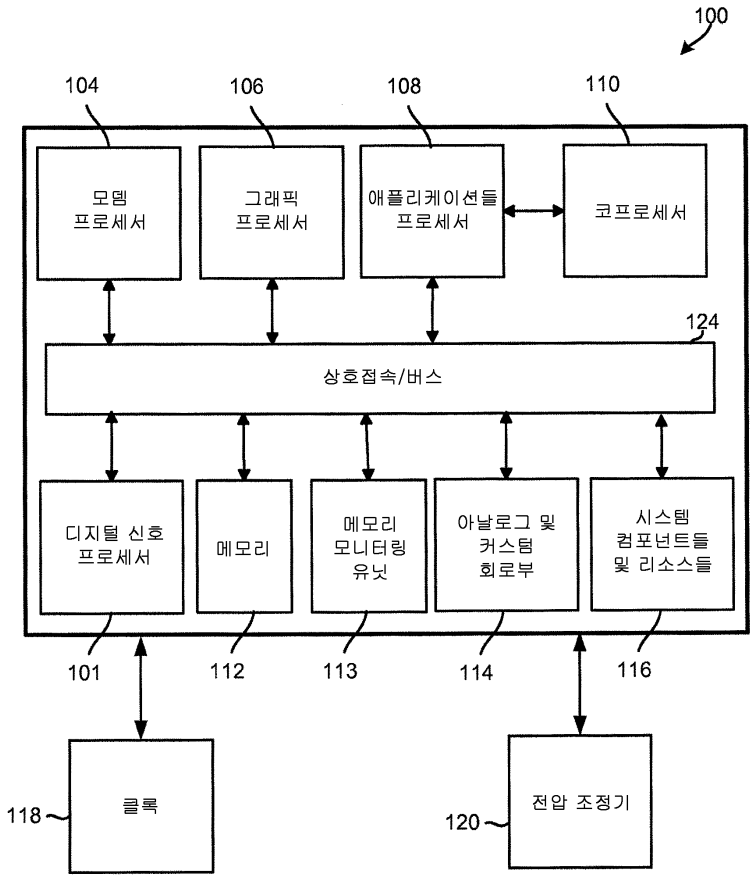
[0151] 본원에 개시된 실시형태들과 관련하여 설명된 다양한 예시적인 로직들, 로직 블록들, 모듈들, 및 회로들을 구현하는 데 사용되는 하드웨어는, 범용 프로세서, 디지털 신호 프로세서 (DSP), 주문형 집적 회로 (ASIC), 필드 프로그래밍가능 게이트 어레이 (FPGA) 또는 다른 프로그램가능형 로직 디바이스, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트, 또는 본원에 설명된 기능들을 수행하도록 설계된 이들의 임의의 조합으로 구현 또는 수행될 수도 있다. 범용 프로세서는 다중프로세서일 수도 있지만, 이와 달리, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한, 컴퓨팅 디바이스들의 조합, 예컨대, DSP 와 다중프로세서의 조합, 복수의 다중프로세서들, DSP 코어와 결합한 하나 이상의 다중프로세서들, 또는 임의의 다른 이러한 구성으로서 구현될 수도 있다. 이와 달리, 일부 단계들 또는 방법들은 소정 기능으로 특정된 회로부에 의해 수행될 수도 있다.

[0152] 하나 이상의 예시적 실시형태들에서, 설명된 기능은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 기능들은 컴퓨터 판독가능 저장 매체 또는 비-일시적 프로세서 판독가능 저장 매체 상의 하나 이상의 프로세서 실행가능 명령들 또는 코드로서 저장될 수도 있다. 본원에 개시된 방법 또는 알고리즘의 단계들은, 비-일시적 컴퓨터 판독가능 또는 프로세서 판독가능 저장 매체 상에 상주할 수도 있는 프로세서 실행가능 소프트웨어 모듈에서 실시될 수도 있다. 비일시적 컴퓨터 판독가능 또는 프로세서 판독가능 저장 매체들은 컴퓨터 또는 프로세서에 의해 액세스될 수도 있는 임의의 저장 매체들일 수도 있다. 제한이 아닌 예로서, 이러한 비일시적 컴퓨터 판독가능 또는 프로세서 판독가능 매체들은, RAM, ROM, EEPROM, 플래시 메모리, CD-ROM 또는 다른 광학 디스크 저장소, 자기 디스크 저장소 또는 다른 자기 저장 디바이스들, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는 데 사용될 수도 있으며 컴퓨터에 의해 액세스될 수도 있는 임의의 다른 매체를 포함할 수도 있다. 본원에 사용된, 디스크 (disk) 및 디스크 (disc) 는 CD (compact disc), 레이저 디스크 (laser disc), 광 디스크 (optical disc), DVD (digital versatile disc), 플로피 디스크 (floppy disk) 및 블루레이 디스크 (blu-ray disc) 를 포함하며, 여기서, 디스크 (disk) 는 보통 데이터를 자기적으로 재생하지만, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 상기의 조합들은 또한, 비일시적 컴퓨터 판독가능 및 프로세서 판독가능 매체들의 범위 내에 포함된다. 또한, 방법 또는 알고리즘의 동작들은 컴퓨터 프로그램 제품에 포함될 수도 있는 비일시적 프로세서 판독가능 매체 및/또는 컴퓨터 판독가능 매체 상의 코드들 및/또는 명령들의 하나 또는 임의의 조합 또는 세트로서 상주할 수도 있다.

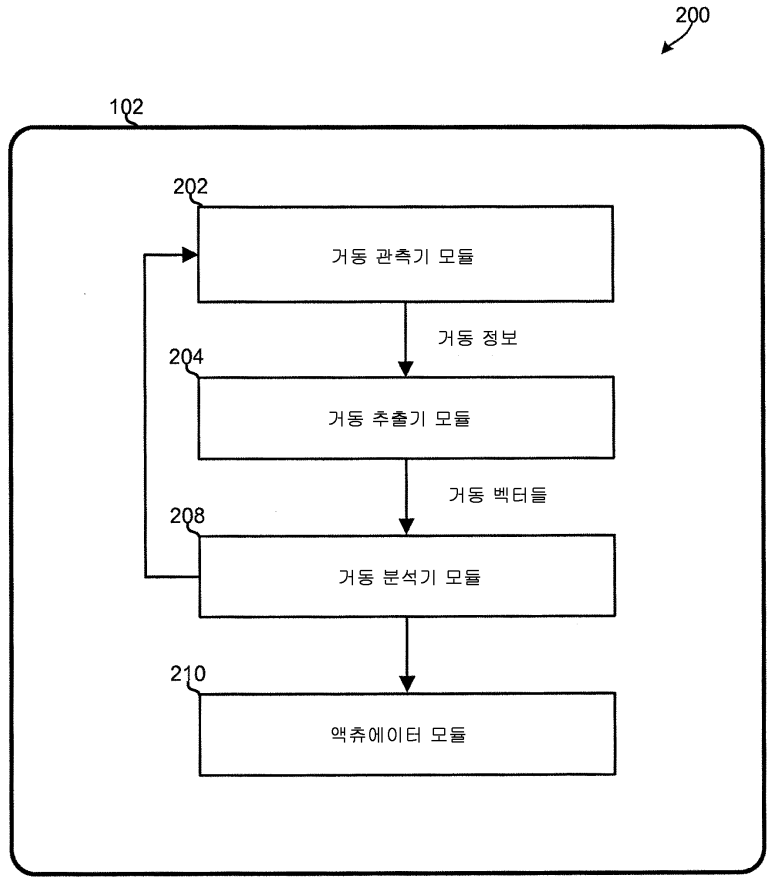
[0153] 본 개시의 실시형태들의 이전 설명은 당업자로 하여금 본 발명을 제조 및 이용할 수 있게 하기 위해 제공된다. 이들 실시형태들에 대한 다양한 수정형태들은 당업자들에게 쉽게 명백할 것이며, 본원에 정의된 포괄적 원리들은 본 발명의 사상 또는 범위로부터 벗어남이 없이 다른 실시형태들에 적용될 수도 있다. 따라서, 본 발명은 본원에 나타낸 실시형태들에 한정되는 것으로 의도되는 것이 아니라, 본원에 개시된 원리들과 신규한 피처들 및 다음의 청구항들에 부합하는 가장 넓은 범위가 부여되어야 한다.

도면

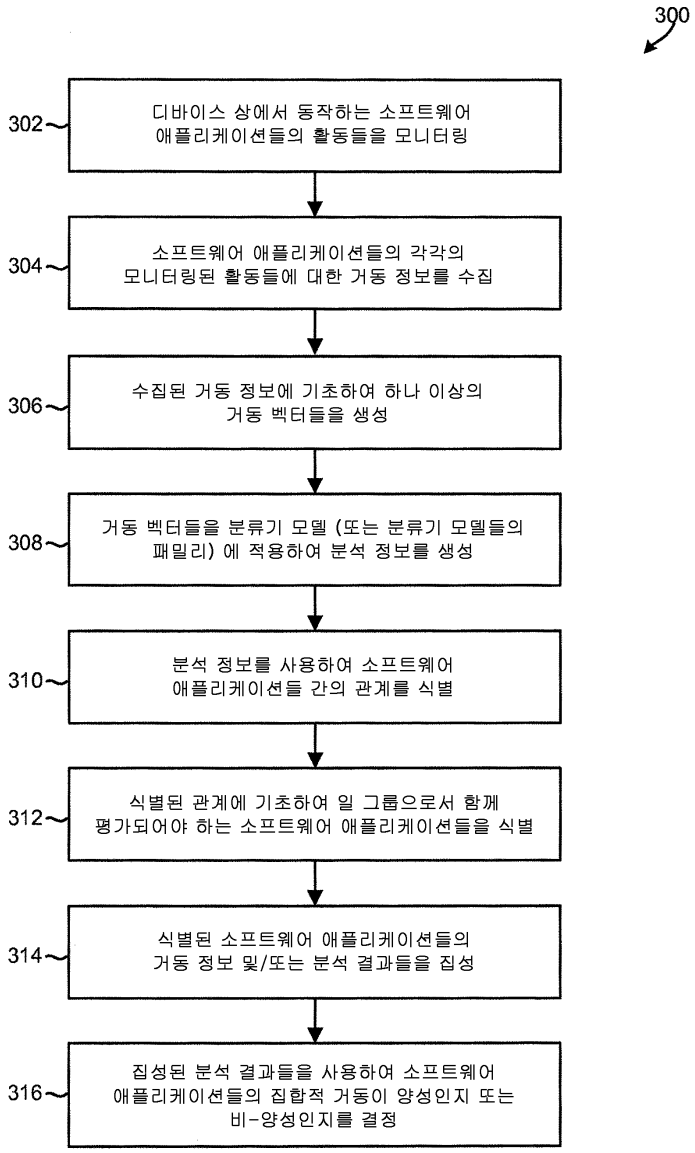
도면1


100
104
106
108
110
모뎀 프로세서
그래픽 프로세서
애플리케이션들 프로세서
코프로세서
124
상호접속/버스
디지털 신호 프로세서
메모리
메모리 모니터링 유닛
아날로그 및 커스텀 회로부
시스템 컴포넌트들 및 리소스들
101
112
113
114
116
118
클록
120
전압 조정기

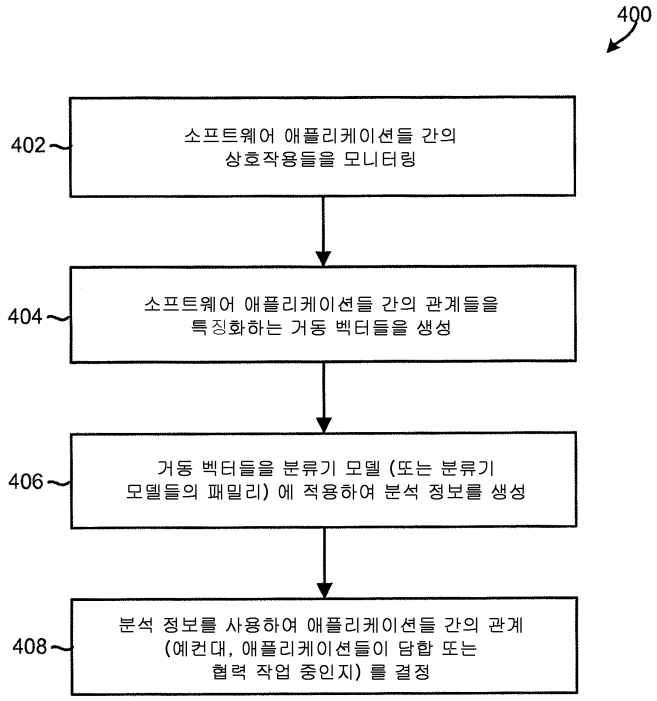
도면2


200
102
202
거동 관측기 모듈
거동 정보
204
거동 추출기 모듈
거동 벡터들
208
거동 분석기 모듈
210
액츄에이터 모듈

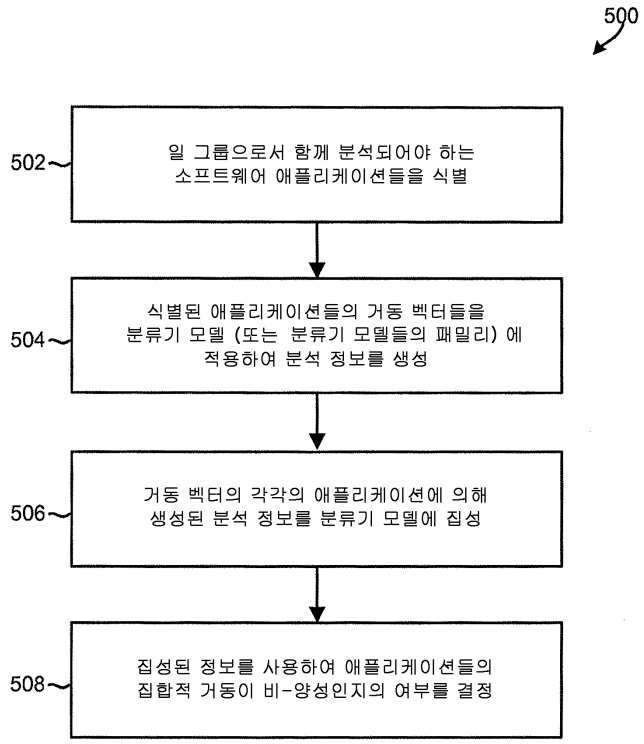
도면3


300
302
디바이스 상에서 동작하는 소프트웨어 애플리케이션들의 활동들을 모니터링
304
소프트웨어 애플리케이션들의 각각의 모니터링된 활동들에 대한 거동 정보를 수집
306
수집된 거동 정보에 기초하여 하나 이상의 거동 벡터들을 생성
308
거동 벡터들을 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용하여 분석 정보를 생성
310
분석 정보를 사용하여 소프트웨어 애플리케이션들 간의 관계를 식별
312
식별된 관계에 기초하여 일 그룹으로서 함께 평가되어야 하는 소프트웨어 애플리케이션들을 식별
314
식별된 소프트웨어 애플리케이션들의 거동 정보 및/또는 분석 결과들을 집성
316
집성된 분석 결과들을 사용하여 소프트웨어 애플리케이션들의 집합적 거동이 양성인지 또는 비-양성인지를 결정

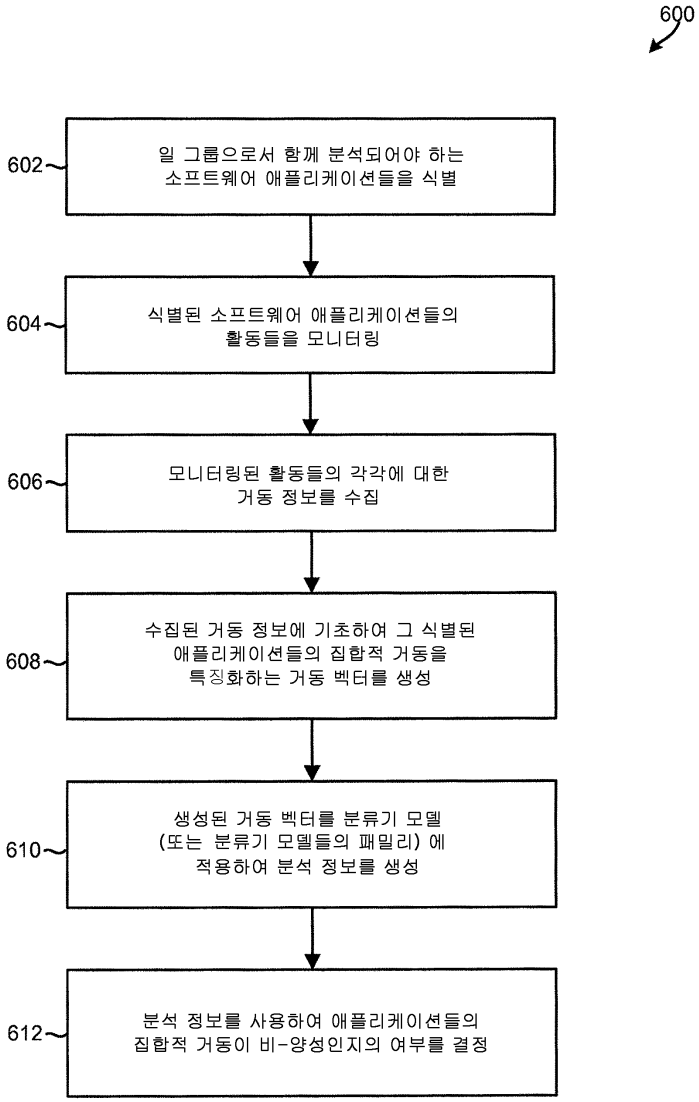
도면4


400
402
소프트웨어 애플리케이션들 간의 상호작용들을 모니터링
404
소프트웨어 애플리케이션들 간의 관계들을 특징화하는 거동 벡터들을 생성
406
거동 벡터들을 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용하여 분석 정보를 생성
408
분석 정보를 사용하여 애플리케이션들 간의 관계 (예컨대, 애플리케이션들이 담합 또는 협력 작업 중인지) 를 결정

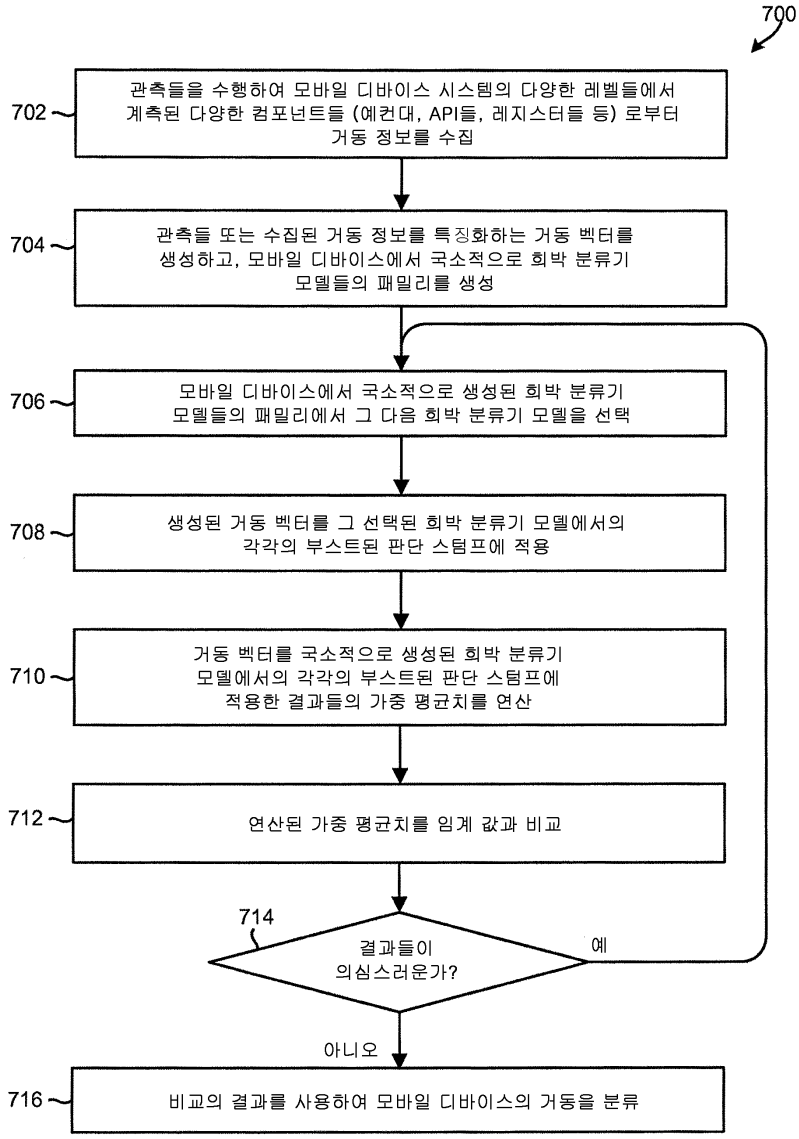
도면5


500
502
일 그룹으로서 함께 분석되어야 하는 소프트웨어 애플리케이션들을 식별
504
식별된 애플리케이션들의 거동 벡터들을 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용하여 분석 정보를 생성
506
거동 벡터의 각각의 애플리케이션에 의해 생성된 분석 정보를 분류기 모델에 집성
508
집성된 정보를 사용하여 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정

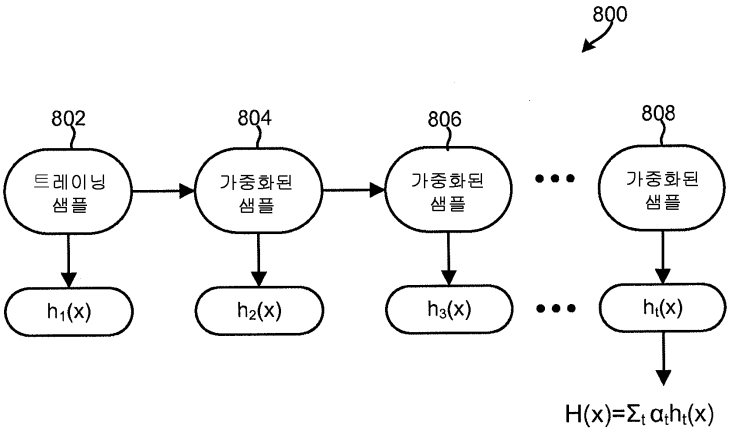
도면6


600
602
일 그룹으로서 함께 분석되어야 하는 소프트웨어 애플리케이션들을 식별
604
식별된 소프트웨어 애플리케이션들의 활동들을 모니터링
606
모니터링된 활동들의 각각에 대한 거동 정보를 수집
608
수집된 거동 정보에 기초하여 그 식별된 애플리케이션들의 집합적 거동을 특징화하는 거동 벡터를 생성
610
생성된 거동 벡터를 분류기 모델 (또는 분류기 모델들의 패밀리) 에 적용하여 분석 정보를 생성
612
분석 정보를 사용하여 애플리케이션들의 집합적 거동이 비-양성인지의 여부를 결정

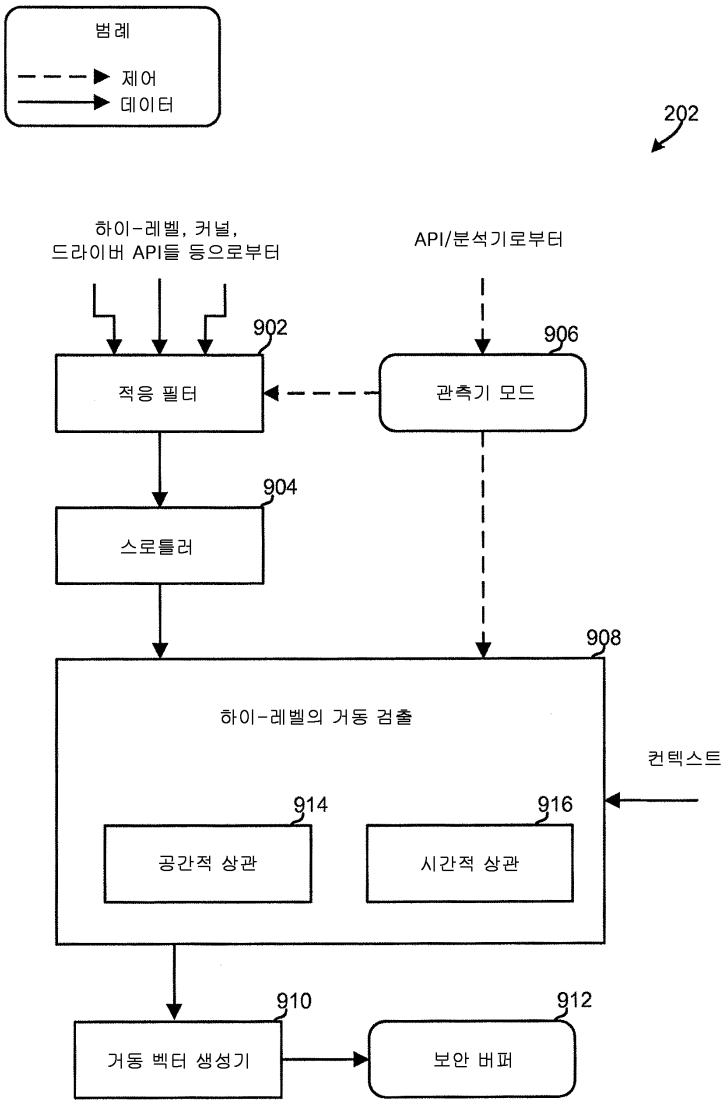
도면7


700
702
관측들을 수행하여 모바일 디바이스 시스템의 다양한 레벨들에서 계측된 다양한 컴포넌트들 (예컨대, API들, 레지스터들 등) 로부터 거동 정보를 수집
704
관측들 또는 수집된 거동 정보를 특징화하는 거동 벡터를 생성하고, 모바일 디바이스에서 국소적으로 희박 분류기 모델들의 패밀리를 생성
706
모바일 디바이스에서 국소적으로 생성된 희박 분류기 모델들의 패밀리에서 그 다음 희박 분류기 모델을 선택
708
생성된 거동 벡터를 그 선택된 희박 분류기 모델에서의 각각의 부스트된 판단 스텀프에 적용
710
거동 벡터를 국소적으로 생성된 희박 분류기 모델에서의 각각의 부스트된 판단 스텀프에 적용한 결과들의 가중 평균치를 연산
712
연산된 가중 평균치를 임계 값과 비교
714
결과들이 의심스러운가?
예
아니오
716
비교의 결과를 사용하여 모바일 디바이스의 거동을 분류

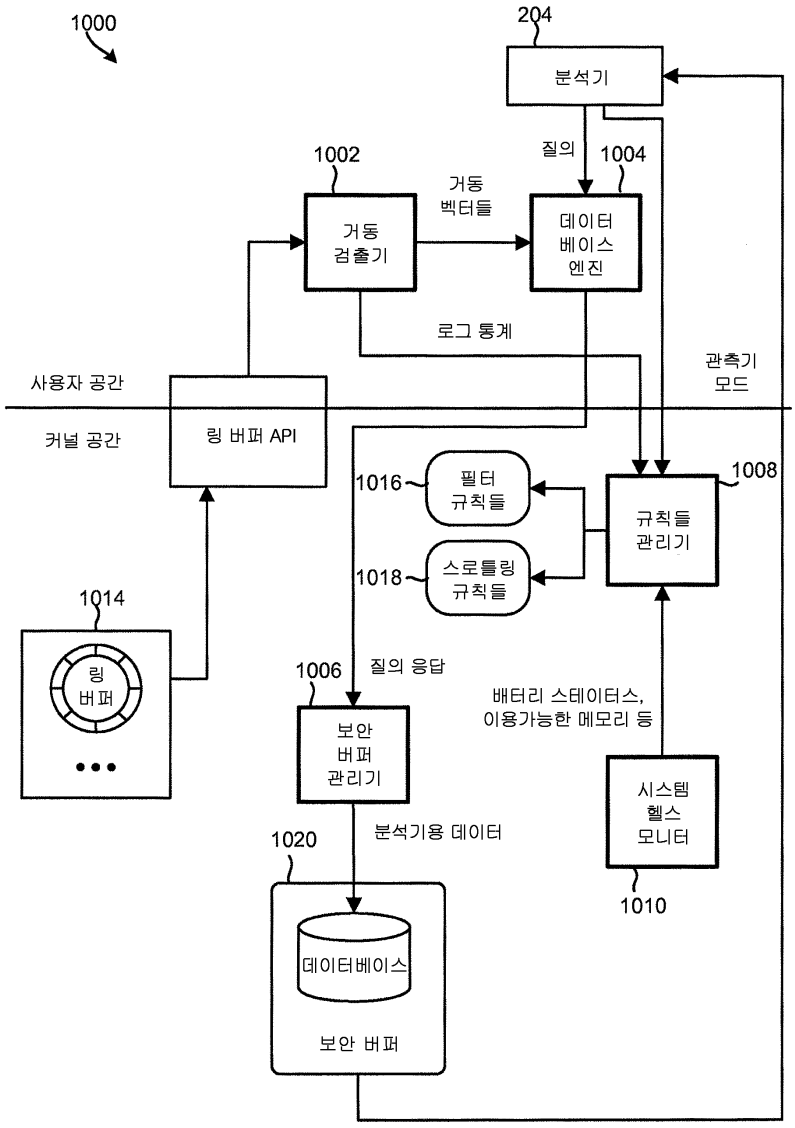
도면8


800
802
804
806
808
트레이닝 샘플
가중화된 샘플
가중화된 샘플
가중화된 샘플
h1(x)
h2(x)
h3(x)
ht(x)
H(x)=Σt αtht(x)

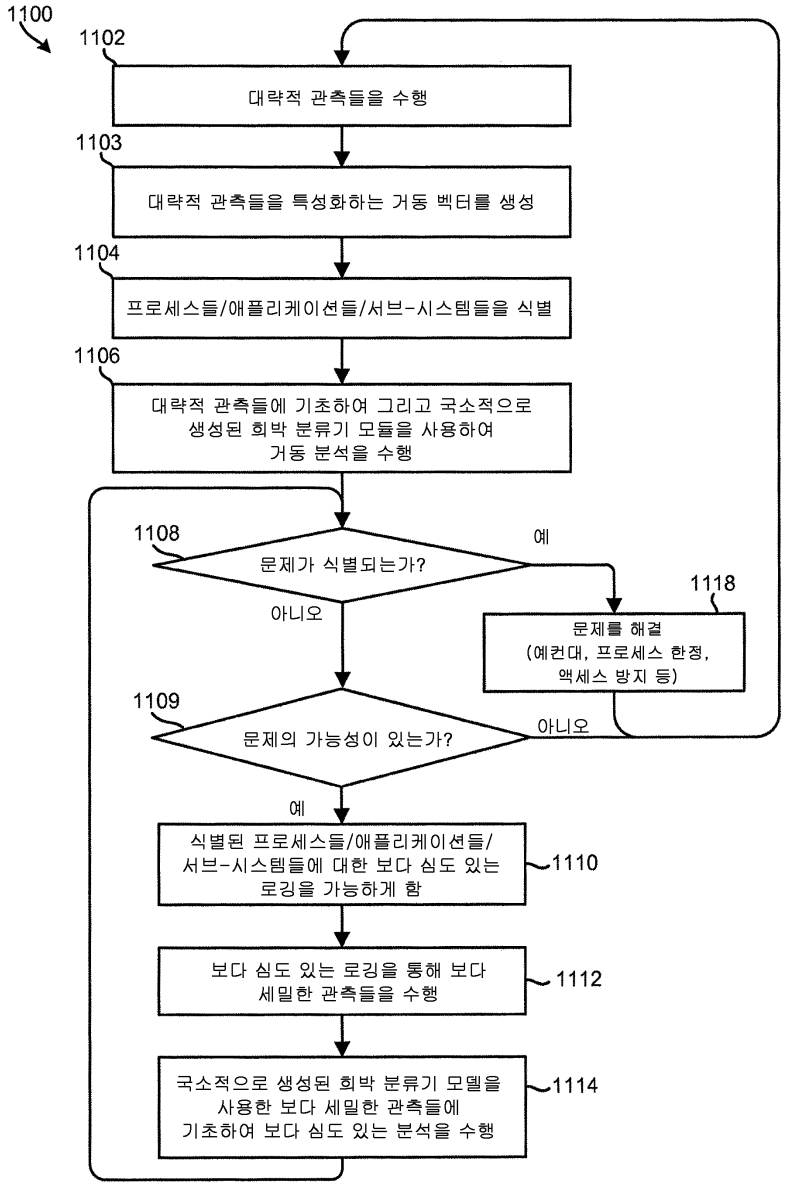
도면9


범례
제어
데이터
202
하이-레벨, 커널,
드라이버 API들 등으로부터
API/분석기로부터
902
906
적응 필터
관측기 모드
904
스로틀러
908
하이-레벨의 거동 검출
컨텍스트
914
916
공간적 상관
시간적 상관
910
912
거동 벡터 생성기
보안 버퍼

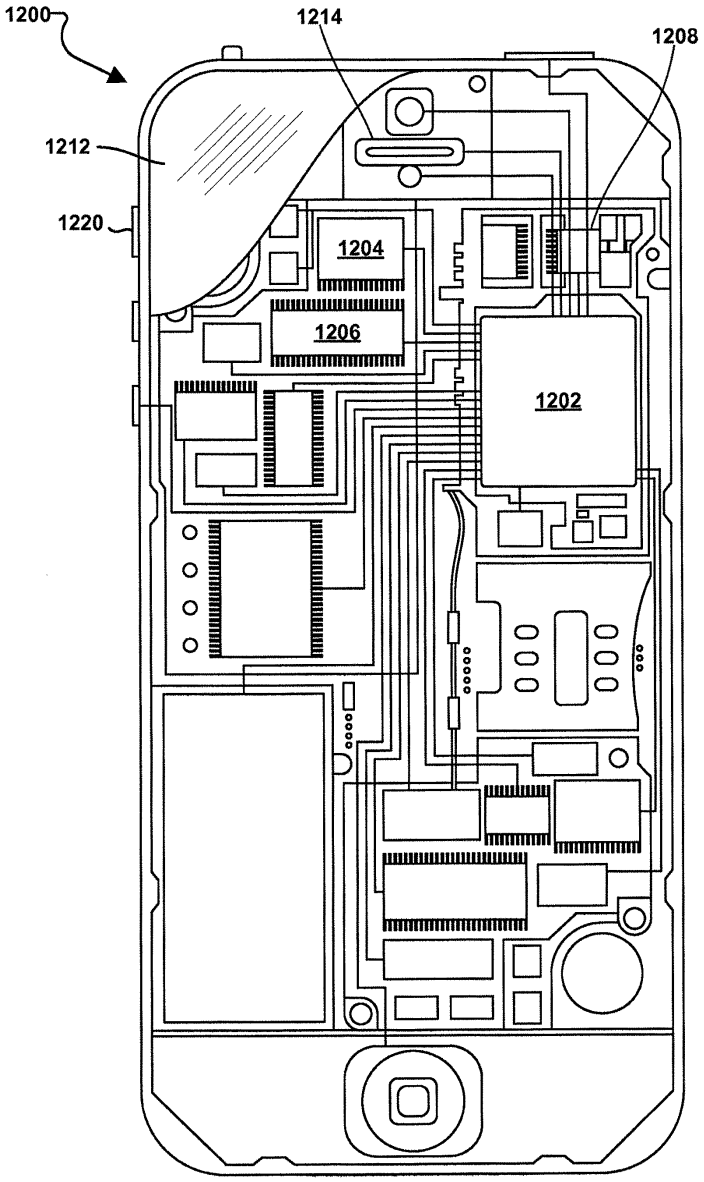
도면10


1000
204
분석기
질의
1004
1002
거동
벡터들
거동
검출기
데이터
베이스
엔진
로그 통계
관측기
모드
사용자 공간
커널 공간
링 버퍼 API
1016
필터
규칙들
1008
규칙들
관리기
1018
스로틀링
규칙들
1014
링
버퍼
1006
질의 응답
배터리 스테이터스,
이용가능한 메모리 등
보안
버퍼
관리기
시스템
헬스
모니터
분석기용 데이터
1020
1010
데이터베이스
보안 버퍼

도면11


1100
1102
대략적 관측들을 수행
1103
대략적 관측들을 특성화하는 거동 벡터를 생성
1104
프로세스들/애플리케이션들/서브-시스템들을 식별
1106
대략적 관측들에 기초하여 그리고 국소적으로 생성된 희박 분류기 모듈을 사용하여 거동 분석을 수행
1108
문제가 식별되는가?
예
아니오
1118
문제를 해결 (예컨대, 프로세스 한정, 액세스 방지 등)
1109
문제의 가능성이 있는가?
아니오
예
식별된 프로세스들/애플리케이션들/서브-시스템들에 대한 보다 심도 있는 로깅을 가능하게 함
1110
보다 심도 있는 로깅을 통해 보다 세밀한 관측들을 수행
1112
국소적으로 생성된 희박 분류기 모델을 사용한 보다 세밀한 관측들에 기초하여 보다 심도 있는 분석을 수행
1114

도면12



도면13