스마트 폰의 터치 스트로크 지속적 인증을 위한 스태킹 커널 릿지 리그레션 네트워크

장인호, 앤드류 테오뱅진
연세대학교 전기전자공학과
e-mail : damonchang23@yonsei.ac.kr

Stacking Kernel Ridge Regression Network for Smart Phone's Touch-Stroke Continuous Authentication

Inho Chang, Andrew Beng-Jin Teoh
Dept. of Electrical and Electronic Engineering, Yonsei University

요 약

이 논문은 스마트 폰에서 터치 스트로크를 이용하여 지속적 인증을 할 수 있는 딥 러닝 네트워크인 스태킹 커널 릿지 리그레션 네트워크 (Stacking Kernel Ridge Regression Network: SKRRN)에 대한 연구이다. SKRRN 은 여러 개의 커널 릿지 리그레션(Kernel Ridge Regression: KRR) 으로 구성되어있고, 계층적이며 모든 KRR 은 해석적이고 독립적으로 훈련된다. SKRRN 은 다른 딥 러닝 네트워크와는 다르게 비가공 터치 스트로크 데이터로부터 특징을 배우지 않고 Hand-Crafted 피처와 같이 추출된 데이터로부터 재학습을 한다. 이러한 재학습은 기존 데이터 셋을 더 구별 하기 쉽고 풍부하게 만들어준다. SKRRN 은 HMOG 데이터 셋을 사용하여 4.295%의 동일 오류율을 달성하였다.

1. 서론

최근 많은 스마트 기기의 보급으로 어디에서든 스마트 폰을 사용 사람을 쉽게 목격할 수 있다. 스마트 폰 사용자들은 보안의 취약함에 노출되어있는 스마트 폰에 아무런 의심 없이 그들의 소중한 정보들을 믿고 저장한다. 이러한 스마트 폰의 보안 인증 시스템은 비밀번호, 패턴, 또는 생체인증 등으로 사용 초입 부분 또는 초입 부분 이후에 한번만 발생한다. 따라서 초입 부분 인증이 끝나면 스마트 기기는 보안 위험에 노출 된다. 이러한 단점을 보완한 기술이 지속적 인증 기술이다[1].

지속적 인증은 사용자의 신원을 기기 백그라운드에서 조용히 감시하고 인증하는 방법이다. 따라서 초입 인증이 끝난 후에도 지속적으로 스마트폰의 보안을 유지 할 수 있다. 지속적 인증은 생체인식을 주로 사용하는데 이러한 생체인식은 크게 신체적 그리고 행위적 생체인식으로 나눌 수 있다. 지문, 홍채, 그리고 얼굴 등 신체적 생체인식이 초입 부분 인증에 사용되는 반면, 지속적 인증에서는 행위적 생체인식인 터치 스트로크가 유망한 차세대 주자로 각광 받고 있다. 비록 터치 스트로크는 신체적 생체인식인 지문, 홍채 등에 비해 분별력이 낮다. 하지만 백그라운드에서 여러 번 지속적으로 인증을 시도하기 때문에 한번의 접촉으로 인증하는 신체적 생체인식처럼 높은 정확도를 가져야 할 필요는 없다.

지속적 인증에 사용되는 터치 스트로크 피처들은 미가공 터치 데이터(손가락 커버 면적, 압력, 그리고 X,Y 좌표 등)에서 추출된다. 추출된 피처들은 보통 hand-crafted 피처들이고 피처 벡터 형식으로 표현된다. 이러한 피처들은 별도의 분류기를 사용하여 실사용자와 침입자를 구별한다. 예를 들어, Support Vector Machine (SVM) 같은 분류기가 주로 사용된다.

딥 러닝은 이미지나 음성과 같이 구조화되지 않은 미가공 데이터를 레이어 방식으로 학습하여 효과적인 특징들을 추출 할 수 있는 능력이 탁월하다[2]. 다양한 영역에서 보편적으로 hand-crafted 데이터가 미가공 데이터보다 많이 사용되어도 hand-crafted 데이터를 이용한 딥 러닝 모델은 많지 않다. 이 논문에서는 커널 릿지 리그레션을 쌓아서 만든 스태킹 커널 릿지 리그레션 네트워크 (Stacking Kernel Ridge Regression Network : SKRRN)를 제안한다. 이 모델은 딥 러닝 모델 중 스태킹 기반 피처 재학습 모델이다. 제안된 모델 안에 모든 커널 릿지 리그레션은 독립적이며 해석학적으로 훈련되어, 딥 러닝에서 깊이가 깊어지면 발생하는 기울기 사라짐 문제는 제안된 모델에서는 발생 하지 않는다. 본 논문은 스마트 폰 지속적 인증을 위해 터치 스트로크 hand-crafted 데이터에 딥 러닝 모델인 SKRRN 을 사용하여 좋은 성능을 보여줄 수 있음을 입증한다.

2. 사전 지식

2.1 릿지 리그레션 (Ridge Regression: RR)

릿지 리그레션은 예측 변수 벡터와 레이블 사이의 선형 함수를 찾는 통계적 문제이다. 리그레션의 해를

찾는 일반적인 방법은 제곱 손실 함수의 합을 최소화 하는 방법인데, 이러한 방법으로 구해진 가중치는 훈련 데이터의 수가 적으면 편차가 심해져서 예측 값을 신뢰하기 힘들어진다. 그 해결책으로 제곱 손실 함수에의 합에 페널티 항을 추가해주어 앞에 언급된 문제를 해결 할 수 있다. 이를 릿지 리그레션이라고 한다.

2.2 커널 릿지 리그레션 (Kernel Ridge Regression: KRR)

릿지 리그레션은 비선형 분류 문제에 맞지 않는 선형 모델이다. 따라서 비선형 모델에 사용하기 위해서 커널 트릭을[3] 사용하면 손쉽게 커널 릿지 리그레션으로 일반화 할 수 있다. 일반적으로 비선형 커널 함수인 다항 함수, RBF 함수 등을 사용한다. 본 논문에서는 RBF 함수를 사용하였다.

3. 방법론

3.1 개요

본 논문에서는 실사용자와 침입자 즉, 2 개의 클래스로 구성된 2 클래스 터치 스트로크 인증 시스템에 초점을 맞추어 연구를 진행하였다. 이 시스템은 안드로이드 API 로부터 하나의 비가공 터치 스트로크 데이터가 기록되면 피처 벡터로 변환 된다. 하지만 하나의 피처 벡터는 적은 양의 실사용자 정보를 담고 있다. 따라서 한 개 이상의 피처 벡터가 수집이 되면 피처 레벨 퓨전[4]이라는 방식을 사용하여 여러 개의 피처 벡터를 한 개의 백터로 만들어 하나의 피처 벡터에 더 많은 정보를 포함 할 수 있게 만든다. 피처 레벨 퓨전에 대한 자세한 내용은 (4.2)에서 설명 할 것이다. 이러한 절차를 거치고 나서 SKRRN 은 합쳐진 피처로부터 재학습을 하고 실험 데이터를 분류한다. (그림 1)에 그 순서도를 나타내었다.

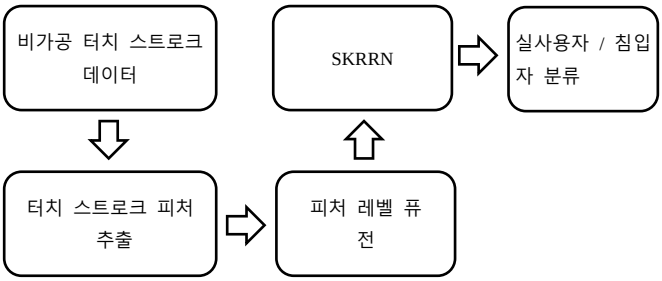

그림 1. 터치 스트로크 지속적 인증시스템 순서도

3.2 스태킹 커널 릿지 리그레션 네트워크

KRR 은 SKRRN 을 구성하는 가장 기본적인 모듈이다. 모든 모듈은 해석학적으로 그리고 독립적으로 훈련이 된다. SKRRN 은 N 개의 KRR 모듈(레이어)로 이루어져 있고 이는 (그림 2)에 나타나있다. SKRRN 은 KRR 모듈의 결과값을 이전 데이터에 추가하여 데이터의 차원을 증가 시킨다. 차원이 증가한 데이터는 다음 모듈의 입력으로 사용되고, 이 절차를 N 번 반복한다. 이렇게 스태킹 된 레이어를 지날수록 추가된 피처들은 증가된 차원과 함께 데이터를 풍부하고 분별력 있게 만들어준다. 각각의 KRR 레이어에서는 Relu 를 추가하여 비선형성을 증가시키는 역할을 했다. 클래스 분류를 위해서는 argmax 를 사용하여 클래스를 분류 하였다.

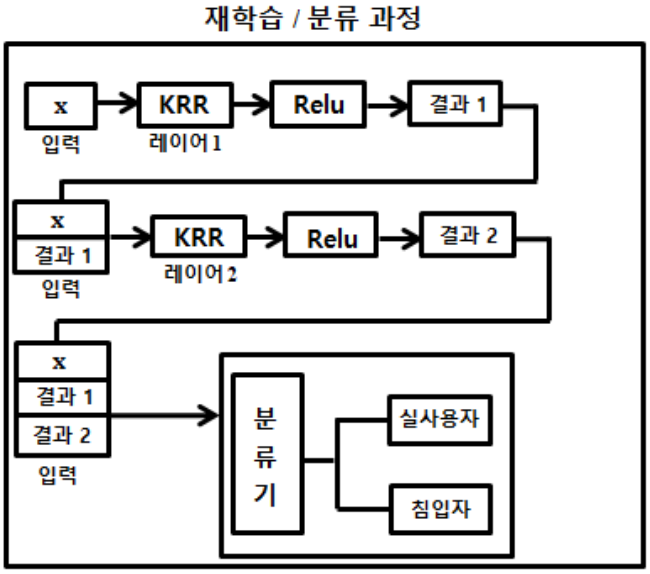

그림 2. N=2, 즉 2 개의 KRR 레이어를 사용했을 때 재학습과 분류 과정을 나타낸 그림. KRR 의 결과값을 원래 입력에 계속 스태킹 하여 재학습을 한다.

4. 실험 및 평가

4.1 데이터 설명

실험에 사용된 데이터는 HMOG 데이터 셋이 사용되었다. 이 데이터 셋은 안드로이드 API 에서 수집되었으며 100 명의 사용자, 17 개의 특성, 그리고 10075 개의 샘플로 구성되어있다. 그러나 100 명의 사용자 중 84 명의 사용자의 데이터만 사용하였다. 그 이유는 몇몇의 유저는 극소량의 데이터만 가지고 있어, 피처 레벨 퓨전을 진행 할 수가 없었기 때문이다. 또한, 이 데이터 셋은 지문읽기와 글쓰기 세션으로 이루어져있는데, 본 실험에서는 터치 스트로크와 관련된 지문읽기 세션 데이터만 사용하였다.

4.2 피처 레벨 퓨전

앞서 언급했던 피처 레벨 퓨전 [4]에 따라 k 개의 연속 샘플을 피처 별로 평균값을 구하여 하나의 피처로써 사용하였다. 즉, k=2 일 경우 2 개의 연속되는 스트로크 피처를 평균 내어 하나의 피처로 만드는 과정이다.

4.3 성능 평가 기준

제안 방법의 성능을 평가하기 위해서 생체인증 시스템에서 가장 많이 사용되는 동일 오류율을 사용하였다. 동일 오류율은 오인식율과 오거부율일 같아지는 비율을 말하는데, 각각의 계산방법은 (식 1, 2)와 같다.

$$\text{오인식율} = \frac{\text{오인식된 데이터의 수}}{\text{전체 침입자 데이터의 수}}(\%) \qquad \text{(식 1)}$$

$$\text{오거부율} = \frac{\text{오거부된 데이터의 수}}{\text{전체 실사용자 데이터의 수}}(\%) \quad \text{(식 2)}$$

본 논문에서는 동일 오류율을 구하기 위해 총 84 명의 유저에 대한 84 개의 SKRRN 이 독립적으로 사용되었다. 각각의 SKRRN 은 실사용자와 침입자의 점수를 결과로 반환한다. 예를 들어, 만약 유저 1 이 실사용자로 지정되어있으면 나머지 83 명의 사용자는 침입자로 설정하고 모델을 훈련 시킨다. 총 모델 84 개의 실사용자, 침입자 스코어를 가지고 총 동일 오류율을 구하였다. 실사용자와 침입자 샘플의 데이터 개수는 피처 레벨 퓨전 변수 k 에 의해 달라진다.

4.4 실험 결과

본 실험은 랜덤 하게 추출한 84 명의 유저 데이터 중 랜덤 하게 추출한 5000 개의 터치 스트로크 데이터를 사용하여 총 5 번에 걸쳐 실험을 반복하였고 동일 오류율(%)의 평균과 표준편차는 (표 1)에 기술하였다. SKRRN 의 파라미터를 조정하기 위해 그리드 서치를 하였다. 훈련과 실험 데이터는 중복 없이 동일한 개수로 구성되어 있다. (그림 3)은 피처 레벨 퓨전변수 k 가 제안된 모델에 미치는 영향을 그래프로 나타내었다. k 가 증가 할수록 모델의 동일 오류율은 감소하지만 현실의 터치 스트로크 인증 시스템에서는 k 가 높을수록 더 많은 양의 스트로크 데이터를 입력받아야 하기 때문에 적절한 수의 k 를 설정 해야 사용자의 편의성과 모델의 성능의 균형을 맞출 수 있다. 고로 본 논문에서는 편의성과 성능의 평행을 맞출 수 있는 k=12 를 사용하였다. (그림 4)는 k=12 일 때, 레이어 개수(N)에 따른 동일 오류율(%) 경향을 보여준다. 동일 오류율은 2 번째 레이어까지 감소하다가 그 이후부터는 다시 증가하는 양상을 보여준다. 이는 제안된 모델이 2 번째 레이어 이후에 오버피팅 되어 이런 현상이 나타남을 알 수 있다. 이는 HMOG 의 데이터가 다른 데이터 셋에 비해 상대적으로 간단한 편이여서 더 깊은 레이어에서 오버피팅이 나는 것을 그림을 보아 알 수 있다.

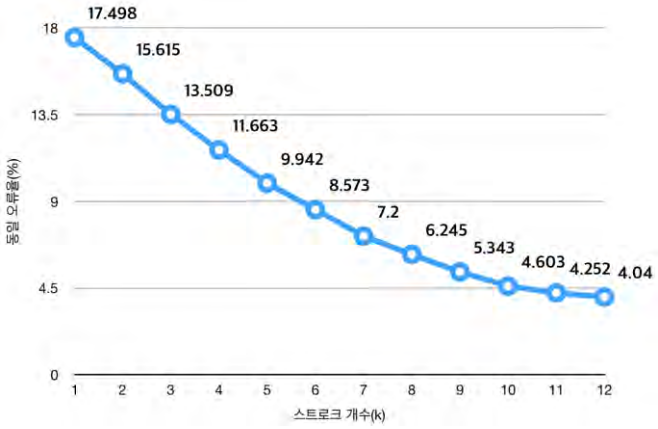

그림 3. 스트로크 개수(k)에 따른 동일 오류율(%) 변화

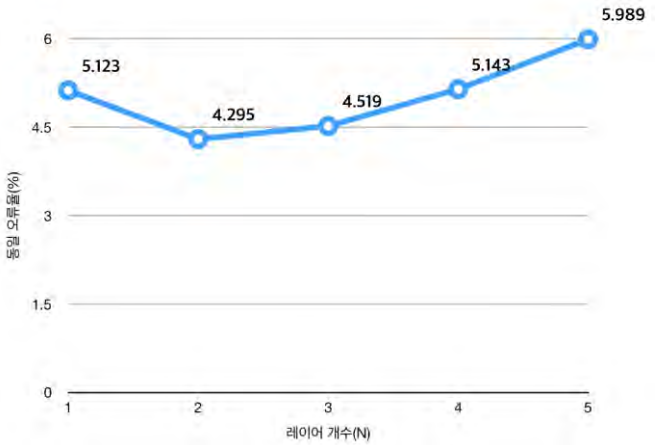

그림 4. 레이어 개수(N)에 따른 동일 오류율(%) 변화

4.5 비교 평가

제안된 모델의 성능비교를 위해 SVM, KRR 과 성능비교를 해보았으며, 모든 모델들은 RBF 커널을 사용하였다. KRR 은 하나의 레이어만 사용한 SKRRN 과 같다. <표 1>은 그 결과를 나타내었다. 모든 실험은 5 번 진행 되어 그 평균을 표기하였다. 그 결과 SKRRN 이 다른 모델들에 비해 좋은 성능을 나타냄을 알 수 있다.

<표 1> KDAN, SVM, KRR 성능 비교

	동일 오류율(표준편차)
KDAN(RBF)	4.2952%(0.53)
KRR(RBF)	7.0067%(0.74)
SVM(RBF)	5.5627%(0.96)

5. 결론

본 논문에서는 스마트 폰에서 터치 스트로크를 이용한 지속적 인증 시스템을 구축하기 위한 스태킹 방식의 피처 재학습 모델에 대해 연구하였다. 이 제안된 모델은 다중의 KRR 모듈로 구성되어 있으며, 각각의 모듈은 해석적으로 그리고 독립적으로 훈련된다. HMOG 데이터 셋을 사용한 결과는 기존의 모델들 보다 더 낮은 동일 오류율을 보여주었다. 다음 연구로써는 기존 2 클래스의 방식으로 진행 하였던 지속적 인증 시스템을 1 클래스 방식으로 SKRRN 을 확장 하는 것 이다.

참고문헌

[1] S. Mondal and P. Bours, “A computational approach to the continuous authentication biometric system,” Information Sciences, vol. 304, pp. 28–53, May 2015.

[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015.

[3] Theodoridis, Sergios (2008). Pattern Recognition. Elsevier B.V. p. 203. ISBN 9780080949123.

[4] Ross, Arun, and Anil Jain. "Information fusion in biometrics." Pattern recognition letters 24.13 (2003): 2115-212.