2005 한국경영과학회/대한산업공학회 춘계공동학술대회
2005년 5월 13일~14일, 충북대학교

GA-SVM Ensemble 모델에서의 accuracy와 diversity를 고려한 feature subset population 선택

성기석, 조성준
서울대학교 공과대학 산업공학과 zoro81@snu.ac.kr , zoon@snu.ac.kr

Abstract

Ensemble에서 feature selection은 각 classifier의 학습할 데이터의 변수를 다르게 하여 diversity를 높이며, 이것은 일반적인 성능향상을 가져온다. Feature selection을 할 때 쓰는 방법 중의 하나가 Genetic Algorithm (GA)이며, GA-SVM은 GA를 기본으로 한 wrapper based feature selection mechanism으로 response model과 keystroke dynamics identity verification model을 만들 때 좋은 성능을 보였다. 하지만 population 안의 후보들 간의 diversity를 보장해주지 못한다는 단점 때문에 classifier들의 accuracy와 diversity의 균형을 맞추기 위한 heuristic parameter setting이 존재하며 이를 조정해야만 하였다. 우리는 GA-SVM 알고리즘을 바탕으로, population안 후보들의 fitness를 측정할 때 accuracy와 diversity 둘 다 고려하는 fitness function을 도입하여 추가적인 classifier 선택 작업을 제거하면서 성능을 유지시키는 방안을 연구하였으며 결과적으로 알고리즘의 복잡성을 줄이면서도 모델의 성능을 유지시켰다.

1. Introduction

Ensemble이란 학습 데이터를 여러 모델에서 학습시킨 후, 이들을 결합하여 최종결과를 얻어내는 기법으로 현재 machine learning, data mining등의 중요한 연구 방향 중 한가지이다[8]. Dietterich[12]는 여러 classifier를 결합하여 최종 결과를 얻는 방법에 대한 연구는 machine learning 연구의 중요한 네 가지 연구 분야 중 하나라고 말하였다. 이러한 ensemble 모델은 각각의 모델의 평균보다 accuracy가 높아지며 각각의 모델로서 얻을 수 없는 일반적인 성능향상을 가져온다[9]. 이러한 ensemble의 성능 향상은 각 classifier들의 각 test data에 대한 예측이 다양한 경우에 성능이 높아지며, 이렇게 예측이 다양한 상황을 diversity가 높다고 한다.

Feature selection이란 모델을 구성할 때 변수들 중에 있는 타겟과 관계가 없는 변수나 변수들 간의 correlation이 높은 변수를 제거하는 기법으로 전통적으로 단일 모델에서 성능을 높이는 데에 쓰였으며, ensemble에서는 accuracy나 학습 시간 등의 특정 성능 측정 방식에 따라 최적의 feature들의 집합을 찾는 과정을 통해 각 classifier의 학습할 데이터의 변수를 다르게 하여 diversity를 높이며, 이것은 상당히 효과적인 방법으로 알려져 있다. 탐색 방식의 특징에 따라 feature selection 알고리즘은 exhaustive search, heuristic search, randomized search로 분류할 수 있으며[7], 이 중 randomized search 방식에서 가장 우위를 차지하고 있는 알고리즘은 다른 알고리즘에 비해 계산 시간이 적게 필요한 GA이다.

FS-Ensemble 모델은 GA를 이용한 wrapper feature selection방식인 GA-SVM과 ensemble을 결합한 모델이며 response model과 keystroke dynamics identity verification model을 만들 때 좋은 성능을 보였다[2],[3]. 하지만 population 안의 후보들 간의 diversity를 보장해주지 못한다는 단점 때문에 classifier들의 accuracy와 diversity의 균형을 맞추기 위한 heuristic parameter setting이 존재하며 이를 조정해야만 하였다

이를 해결하기 위해 이 논문에서는 GA-SVM 알고리즘을 바탕으로, population 내 후보들의 fitness를 측정할 때 Non- dominated Sorting Genetic Algorithm[4]에서 쓰인 sharing 함수를 적용하여 population set의 diversity를 정량화하여 accuracy와 diversity 둘 다 고려하는 fitness function을 만든다. 이를 통해 추가적인 Classifier-Selection method를 제거할 수 있으며 모델의 복잡성을 감소시킬 수 있었다.

본 논문은 6개의 장으로 구성되어 있으며, 다음 장에서는 ensemble과 feature selection의 literature review를 소개하고 3장에서 FS-Ensemble 모델의 개선 방안에 대해 설명을 기술하였다. 그리고 4장에서 실험 데이터에 대해 소개를 하고 5장에선 결과를 요약하였다. 마지막 6장에서는 결론과 앞으로의 연구 과제를 요약한다.

2. Literature Review

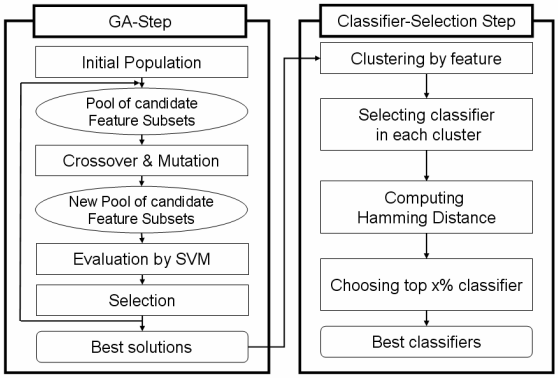
2.1 Ensemble

Ensemble을 향상시키기 위한 연구는 크게 세 가지로 분류되는데[6] 첫 번째로, Classifier들마다 초기 weight를 다르게 하는 방법은 neural network에서 output weight vector와 성능의 상관관계를 알아본 Sharky[17]의 논문 등이 있었으나 성능과 weight값의 변화에는 큰 상관 관계가 없는 것으로

평가되었다. 두 번째로, classifier들의 모델이나 data를 다르게 주어 학습의 방식과 영역을 다르게 하는 방법 중에는 각 classifier의data를 바꾸어 주는 방식이 가장 널리 쓰인다. 이 방식은 다시 data의 레코드를 classifier들마다 달리하는 pattern selection 방식과 학습에 참여하는 data의 변수를 달리하는 feature selection방식으로 나뉘게 된다. Pattern selection방식에는 Bagging[11], Boosting[13] 기법이 있으며, feature selection방식에는 Opitz[5], Sullivan[14], 그리고 FS- Ensemble[2],[3] 등의 연구가 있었다. 마지막으로, 학습을 할 때 학습 영역에서의 탐지 방법의 변화를 주는 방법은 penalty term을 error함수에 사용한 Negative Correlation Learning [15], GA를 이용한 Yao[16]의 연구 등이 있었다.

2.2 FS-Ensemble

FS-Ensemble은 Yu[2]가 제안한 feature selection 방식으로 GA를 이용한 wrapper



<Figure 1> FS-Ensemble

feature selection방식인 GA-SVM과 ensemble을 결합한 모델이다. FS-Ensemble은 크게 GA-SVM 단계와 Classifier-Selection 단계로 나누어진다.

GA-SVM 단계에서는 GA를 이용할 때, population을 base learner인 SVM(Support Vector Machine) 으로 평가하여 다음 세대의 population을 구하는 feature subset selection 을 실행한다[2]. Wrapper approach 방식인 이
기법이 SVM을 base learner로 사용하는 이유는 SVM이 neural network의 성능과 비교하여 비슷한 성능을 보이면서도 훨씬 빠른 학습시간을 보이기 때문이다[2]. 초기 population은 무작위로 선출된 binary string으로 표현되는 후보들을 만들게 되는데 이는 어떤 변수가 선택 되었는지를 나타낸다. 이후 후보들은 GA에 의해 crossover와 mutation을 거친 뒤 fitness function에 의해 평가된 후 이 값에 따라 다음 세대를 선출하게 된다. Population의 크기가 M이면, crossover rate에 따라 offspring이 생겨나게 되고 이것들은 부모의 자리를 대신 차지하게 된다. 그리고 M개의binary string을 fitness 값에 따라 확률적으로 하나씩 뽑는 과정을 M번 반복하여 새로운 세대를 구성하게 된다. 이 과정을 지정한 횟수만큼 반복하고 나서 최종 feature set의 후보들이 나오게 된다. Yu[2]의 논문에서는 fitness를 다음과 같이 산출하며, 공식은 다음과 같다.

$$\text{Fitness}(x)=\alpha\,\text{Acc}(x)+\beta\frac{1}{\text{LrnT}(x)}+\gamma\frac{1}{\text{DimRat}(x)} \quad (1)$$

이때, fitness(x)는 feature subset x에 의해서 계산되는 적합도를 측정한 값이다. Acc(x)는 feature subset x를 이용해 test를 하고 나서의 accuracy, LrnT(x)는 SVM을 학습하는 동안 걸린 시간, DimRat(x)는 전체 feature의 개수에 대해 얼마나 feature subset x의 feature개수가 줄어들었는지를 계산한 것이다. Yu의 논문[2],[3]에서는 α, β, γ를 각각 10, 1/100, 1로 정해 놓고 실험을 하였다.

Classifier-Selection 단계에서는 앞에서 설명한 GA-SVM 단계에서 방법을 사용한 후의 결과물인 M개의 feature subset의 후보들 중에서 실제 ensemble에 참여할 후보를 선정하는 단계이다. GA가 진행되는 동안 초기에는 feature subset의 후보들은 높은 diversity를 보이고 있으나 전체적으로 낮은 accuracy를 보인다. GA가 계속 진행되면 feature subset의 후보들은 전체적으로accuracy가 높아지나 diversity는 감소하는 경향을 보인다. ensemble의 성능을 높이기 위해서는 전체 후보들이 diversity를 가진 채로 accuracy가 높아야 하기 때문에 이들 간의 trade-off가 필요하다[3]. Diversity와 accuracy간의 trade-off를 고려하면서 ensemble을 만들기 위해 Yu가 제안한 Classifier-Selection의 과정은 <Figure 2>와 같다. 가장 diversity를 높게 하는 classifier들의 집합 을 찾는 작업은 NP-complete로 알려져 있다[10]. Yu의 논문에서는 그렇기 때문에 이와 같은 heuristic 방법을
썼다.

FS-Ensemble Procedure

GA-Step

Step 1. Evolve a population of classifiers each of which employs a subset of features for $(1-\varepsilon)\cdot 100\%$ of time T, where T is the time necessary for convergence and $0<\varepsilon<1$.

Step 2. Return all those "premature" classifiers.

Classifier-Selection Step

Step 1. Cluster SVM classifiers f_i's based on the set of features employed. i.e., classifiers in a same cluster use the same subset of features. The SVM parameters of these classifiers such as γ and cost c are different.

Step 2. In each cluster, select one classifier f_i that has the smallest validation error, resulting in a total of $k(\le n)$ classifiers.

Step 3. For each classifier f_i, compute validation output vector $\vec{f}_i = (f_i(1), f_i(2), \ldots, f_i(N))$, where $f_i(k)$ is the output of classifier i for the j^{th} validation pattern.

Step 4. Compute hamming distance $HD(i,j)$ between every pair of validation output vectors $\vec{f}_i$ and $\vec{f}_j$. (*There are a total of ${}_kC_2$ such distances.)

Step 5. Choose the top $x\%$ of the HDs and return the classifiers involved. (*For instance, if $HD(3,7)$ was chosen, classifiers 3 and 7 are identified and returned.)

<Figure 2> FS-Ensemble Creation Method

이러한 FS-Ensemble 알고리즘의 문제점은 크게 두 가지이다. 첫째, 고려해야 할 parameter가 많다는 점이다. GA-SVM 단계가 지난 뒤에 ε 값을 선정해야 하며, Classifier-Selection 단계에서는 classifier중에서 validation data를 예측할 때 각기 다르게 성능을 예측한 classifier 집합을 산출하는 작업에서 x값을 선정해야 한다. 둘째, 이러한 parameter 선정을 포함한 Classifier-Selection 단계에서의 알고리즘이 복잡하다는 점이다. 이러한 복잡

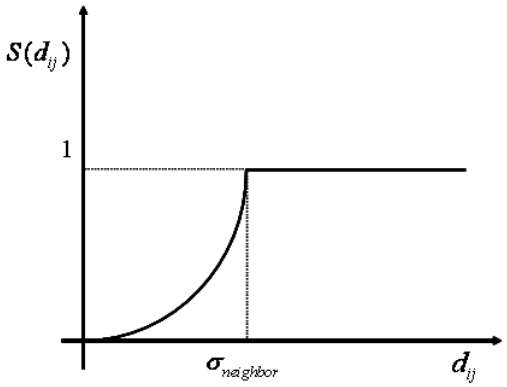
성의 증가는 알고리즘 구현의 어려움을 가져올 수 있다. 그러므로 우리는 고려해야 할 parameter의 개수를 줄이고, Classifier- Selection 단계의 알고리즘 단순화를 목표로 하면서 알고리즘의 성능은 유지시키는 방안을 연구한다.

3. Proposed Method

FS-Ensemble 알고리즘에서 Classifier-Selection 단계를 제거하기 위해 GA-SVM 단계에서 classifier들 간의 diversity를 미리 고려할 수 있는 조건식을 추가 하였다. Diversity는 각 classifier들의 test data에 대한 예측이 다양한 경우에 상승하게 된다. 하지만 우리가 test set을 미리 알지 못하므로 Opitz[5]의 연구처럼 validation set에 대한 예측의 다양성을 토대로 diversity를 측정하고 그 조건식을 fitness function에 추가할 수 있다. 하지만 이 방법은 GA과정에서 search 영역의 다양화를 보장하지는 못한다. 그래서 우리는 fitness function 안에 새로운 term, degree of uniqueness를 넣어서 실험을 하기로 하였다. 이 조건식은 feature subset들이 입력 영역 안에서 서로간의 떨어져 있는 정도를 hamming distance를 이용하여 계산한 것으로 Srinivas의 논문[4]의 Sharing function을 변형하여 이용하였다. Sharing function은 Srinivas의 논문에서 GA알고리즘에서 탐색을 더 잘 할 수 있게 하는 목적으로 population의 한 후보가 다른 후보간의 거리가 멀리 있을수록 다음 세대에 선택될 확률을 높이기 위해 fitness값을 계산할 때 사용하는 함수이다. 이 논문에서는 Sharing function을 응용하여 FS-Ensemble에서 accuracy와 diversity를 고려하는 degree of uniqueness 조건식을 만들었으며 population안의 두 후보간의 거리 Sdistance를 다음과 같이 정의한다.

$$S(d_{ij}) = \begin{cases} (\frac{d_{ij}}{\sigma_{neighbor}})^2, \text{ if } d_{ij} < \sigma_{neighbor}; \\ 1, \qquad \text{otherwise.} \end{cases} \qquad (2)$$

위의 식에서 d_{ij} 는 population 후보 i와 j간의 hamming distance이고, $\sigma_{neighbor}$ 는 두 후보간의 이웃이라고 인정하는 최대 hamming distance이다. 이것은 d_{ij} 의 최대 값인 데이터 패턴 차원의 비율로 계산된다. 그러므로 후보간의 hamming distance가 가까울수록 Sdistance값은 작아지게 된다. Sdistance값을 계산할 때 최대거리를 1로 한정하고 단순히 스케일링을 통해 0에서 1사이로 조정을 하지 않는 이유는 일반적인 population 위치와 멀리 떨어진 후보와 다른 후보들 간의 거리가 클 수 있기 때문에 이것으로 인한 Sdistance의 분포가 왜곡되는 것을 방지하기 위해서이다. 이때 이 값이 다음 계산 과정인 Uniqueness 계산에 많은 영향을 끼칠 수 있다. 또한 Sdistance값을 계산할 때 단순히 거리를 계산하게 되면 일반적인 population 위치와 멀리 떨어진 outlier 후보들에게 이득이 될 수 있기 때문에 어느 이상 거리가 떨어진 경우 1로 지정한다. 또한, hamming distance를 $\sigma_{neighbor}$ 로 나누어 제곱을 하는 이유는 거리가 가까울수록 Sdistance값을 작게 하여 거리가 멀수록 이득을 주기 위함이다. 위 함수의 그래프는 다음과 같다.



<Figure 3> Sdistance 함수의 그래프

이때, degree of uniqueness값은 다음과 같다.

$$U(k) = \frac{\sum_{k \neq j} S(d_{kj})}{n-1} \qquad (3)$$

예를 들어 4개의 3차원 feature set이 다음과 같이 있다고 하자.

Point1 : 1 1 1 Point2 : 1 1 0
Point3 : 0 0 0 Point4 : 1 0 1

이때 $\sigma_{neighbor}$ = d_{max} X 2/3 = 2라고 설정할 경우 Sdistance의 값은 다음과 같다.

$$S(d_{ij}) = \begin{pmatrix} 0 & \frac{1}{4} & 1 & \frac{1}{4} \\ \frac{1}{4} & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ \frac{1}{4} & 1 & 1 & 0 \end{pmatrix} \qquad (4)$$

그리고 degree of uniqueness값은 다음과 같다.
U(1) = 0.5, U(2) = 0.75,
U(3) = 1, U(4) = 0.75

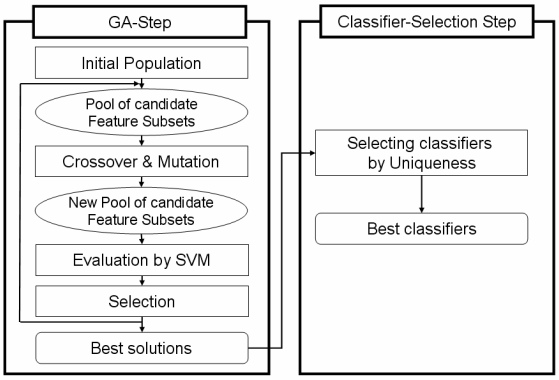
Sdistance는 population의 후보들 간의 input vector상에서 거리가 얼마인지에 대해 정의한 것이고 degree of uniqueness는 population의 한 후보가 population의 다른 후보들과 input vector상에서 얼마나 거리가 떨어져 있는지 알려주는 값이다. n-1로 나눈 이유는 population의 수에 관계없이 0에서 1사이의 값을 갖도록 하기 위해서이며, 주위에 후보들이 많이 있을수록 0에 가까워지게 되고, 다른 후보들과 많이 떨어진 경우 1에 가까운 값을 가지게 된다. 이와 같이 degree of uniqueness를 정의하는 이유는 GA과정에서 population의 input vector간의 거리를 측정하여 이를 유지해 나가는 방식으로 최종 population의 diversity를 높이기 위함이다.

우리는 위의 degree of uniqueness함수를 이용하여 fitness function의 식을 다음과 같이 정의하였다.

$$\text{Fitness(x)}=\alpha\text{Acc(x)}+\beta\frac{1}{\text{LrnT(x)}}+\gamma\frac{1}{\text{DimRat(x)}}+\delta U(x) \quad (5)$$

Fitness 함수에 criteria로 들어가는 learning time과 dimension reduction rate는 population간의 차이가 크지 않으므로 fitness 값의 간결함을 위해서 제거될 변수로 선정되었다 ($\beta=0, \gamma=0$). learning time, dimension reduction rate를 제거하고 그 대신에 degree of uniqueness criteria를 첨가하였다. 그리고 식의 간결함을 위해 α,δ의 비율을 1로 설정하였다.

$$\text{Fitness(x)}=\text{Acc(x)}+U(x) \quad (6)$$



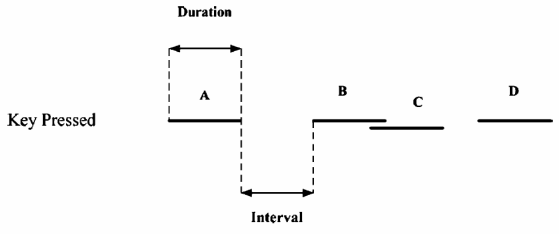
<Figure 4> Proposed FS-Ensemble model

4. Experimental Design

4.1 Data

Keystroke dynamics identity verification model은 기존의 암호 방법의 단점을 보완하고 보안을 강화할 수 있는 방법으로 암호 타자 패턴을 이용한 사용자 인증 방법이다[1]. 암호 패턴은 timing vector로서, 각 문자의 해당키를 누르고 있는 시간(keystroke duration time)과 앞 키가 눌러진 상태에서 다시 올라온 시간(release time)과 다음 키가 눌러진 시간(press time)과의 차이(keystroke interval time)으로 정의된다.

데이터 수집은 Sun Sparc workstation 상에서 이루어졌으며, 시간 측정은 X window system을 이용하였다. 21명의 사용자의 데이터를 가지고 각각의 데이터에 대하여 여러 가지 알고리즘에 대해 비교실험을 하였다. 주인패턴에서 뒷부분 75개는 테스트용으로, 나머지는 학습용으로 사용한다. 주인패턴의 학습패턴에 대해 전처리를 행하여 주인의 일반적인 타자 특성을 벗어나는 패턴은 제거하였다.



<Figure 5> "ABCD"암호의 keystroke dynamics

타인은 총 15명의 실험데이터를 얻었다. 타인들은 각 주인공에 대해 5번씩의 암호 타자를 하였다. 그래서 각 주인공 당 총 75개씩의 타인 패턴을 얻었다. 이때 타인들에게 암호에 대해 학습을 하여 실제 상황과 비슷하게 만들기 위해 타인 그룹은 주인공의 암호가 한글을 바탕으로 조어한 것이라면 그 정보를 피실험자에게 제공하고 암호를 입력하게 하였다.

4.2 Model setting

Yu의 논문[3]에서 쓰인 학습용 패턴 데이터와 여기에서 쓰인 학습용 패턴 데이터는 개수는 50개로 같지만 random sampling을 한 것이기 때문에 학습 데이터 자체가 달라 직접 Yu의 실험결과와 비교하기가 어려운 점이 있었다. 그래서 여러 모델들을 직접 구현하여 실험을 하기로 하였다. 모델들은 다음과 같다.

▪**Single-FF:** Single SVM with Full Feature Set
▪**Single-UniFit:** Single SVM with Feature Selection with fitness function,

$$\text{Fitness(x)}=\text{Acc(x)}+U(x)$$

▪**Single-YuFit:** Single SVM with Feature Selection with fitness function,

$$\text{Fitness(x)}=10\text{Acc(x)}+\frac{1}{100}\frac{1}{\text{LrnT(x)}}+\frac{1}{\text{DimRat(x)}}$$

▪**Ensem-UniFit-Fix**: SVM Ensemble based on Feature Selection with fitness function, $\text{Fitness(x)}=\text{Acc(x)}+U(x)$ and fixed number of classifier selection by fitness function

아이디	암호	학습패턴수	차원	전처리 제거율 (%)	한글 비율(%)
atom	loveis,	207	15	21	0
celavie	i love 3	330	17	15	0
crapas	autumnman	111	19	10	0
daeguri	90200jdg	164	17	10	0
gmother	rla sua	101	17	18	38
gusegi	dhfpql,	232	15	8	86
jmin	love wid	101	17	19	38
june	dltjdgml	151	17	14	100
oscar	dusru427	365	17	27	63
perfect	manseiii	86	17	25	0
silee	rhkdwo	205	13	20	100
wooks	beaupowe	76	17	24	0
yanwenry	tmdwnsl1	108	17	18	88
yuhwa	yuhwa1kk	388	17	12	0
bubugi	anehwksu	319	17	10	100
dry	tiddmswid	337	19	10	100
jywoo	drizzle	299	15	10	0
megadeth	dlfis wp	342	17	6	88
orange	c,s,93/ksy	200	21	22	0
shlee	dirdhfmw	309	17	33	100
ysoya	ahrfus88	260	17	20	75

<Table 1> 21명의 암호와 학습 패턴 데이터

▪**Ensem-YuFit-Select**: SVM Ensemble based on Feature Selection with fitness function, $\text{Fitness(x)}=10\text{Acc(x)}+\frac{1}{100}\frac{1}{\text{LrnT(x)}}+\frac{1}{\text{DimRat(x)}}$ and classifier-Selection Method by Yu

세 가지는 single 모델이고 나머지 두 가지는 ensemble 모델이다. Ensem-UniFit-Fix 모델은 우리가 제안하는 모델로서, Yu의 논문[3]에서 제안한 방법을 사용한 모델인 Ensem-YuFit-Select 와 다른 점은 fitness function이 다르고 ensemble을 할 때

classifier를 선택하는 과정을 우리가 제안하는 모델은 Yu의 방법을 사용하지 않는다. 이는 이 과정이 population의 diversity를 유지하기 위한 방편으로 개발된 classifier selection 방법이기 때문에 우리는 목적식 값이 높은 순서대로 최종 후보들 중 일정 수를 뽑아 ensemble의 멤버로 참여시켰기 때문이다. 각 classifier의 ɣ ,υ 는 ensemble의 성능을 향상시키기 위해 여러 번의 실험을 통하여 확인된 일정 accuracy 이상을 보이는 범위 안에서 랜덤하게 설정하였다. 학습 시에는 패턴 중에서 50개를 학습에 쓰고 실험을 하였다. 그 50개 패턴 중에서 35개는 training에 쓰고 나머지 15개는 fitness function을 계산할 때 쓰일 validation으로 할당하였다. GA-SVM과정에서 fitness function에 쓰인 accuracy와 learning time은 Gaussian kernel을 사용한 SVM에 의해 측정되었으며, 이때 SVM학습과정에 쓰인 ɣ, cost, ,υ 는 여러 번의 실험을 통해 일반적인 학습 성능이 높은 값을 찾아서 고정하여 학습하였다. GA과정에서는 population size는 100, generation은 50으로 설정하였다. 이는 <Table 2>의 population과 generation에 따른 최종 멤버로 뽑힌 classifier들의 fitness 값을 토대로 population 크기와 generation을 줄이면서도 성능에 크게 영향을 안 주는 population과 generation을 선정하였다. Single -UniFit 모델과 Single-YuFit은 GA- SVM 학습후 가장 각각의 fitness값이 높은 feature를 이용하여 single SVM 성능을 측정하였다. 모든 실험은 다섯 번의 반복 실험 후 평균을 낸 값이며, 두 ensemble 모델의 setting은 <Table 3>과 같다.

Ensem-UniFit-Fix			
	population		
generation	100	200	300
50	1.60	1.67	1.65
100	1.58	1.66	1.67
200	1.61	1.62	1.68
평균	1.60	1.65	1.66

Ensem-YuFit-Select			
	population		
generation	100	200	300
50	22.04	22.09	21.57
100	22.89	22.56	22.26
200	23.00	22.74	22.71
평균	22.64	22.46	22.18

<Table 2> population과 generation에 따른 fitness 값 비교

Setting	Model	
	Ensem-UniFit-Fix	Ensem-YuFit-Select
Fitness function	Fitness = FRR(x) + U(x)	Fitness = 10FRR(x)+ 1/(100 * LrnT(x)) + 1/DimRat(x)
population	100	100
generation	50	50
crossover rate	0,3	0,3
mutation rate	0,01	0,01
degree of neighbor	0,2	
early stoping criterion		0,2
classifier HD percentage		0,3

<Table 3> ensemble model setting

5. Result

먼저 전체 모델들의 결과는 <Table 4>, <Table 5>과 같다. 표에서 각 항목은 다음과 같다.

▪Accuracy : test set 150개 패턴중 예측이 맞은 비율

▪FAR : False accept rate. test set 중에 타인패턴을 본인패턴으로 분류한 비율

▪FRR : False rejection rate. test set 중에 본인패턴을 타인패턴으로 분류한 비율

▪Average uniqueness : 최종 ensemble에 참여하는 후보들의 uniqueness의 평균

▪Num of Ensemble : Ensemble모델의 classifier의 개수

Ensemble 모델에서의 average uniqueness차이가 상당히 많이 난다는 것을 알 수 있다. 우리가 제안하는 Ensem-UniFit-Fix 모델의 average uniqueness는 대부분의 데이터 셋에서 0.76에 가까운 값을 볼 수 있었고 Ensem- YuFit-Select모델에서는 0.22정도의 값을 갖는

Password	Models									
	Ensem-UniFit-Fix Fitness = Acc(x) + U(x)					Ensem-YuFit-Select Fitness = 10Acc(x) + 1/(100 * LrnT(x)) + 1/DimRat(x)				
	Ensemble Accuracy	FAR	FRR	Average Uniqueness	Num of ensemble	Ensemble Accuracy	FAR	FRR	Average Uniqueness	Num of ensemble
90200idg	52,80	24,00	70,40	0,72	31	55,73	69,06	19,46	0,17	10,80
ahrfus88	89,60	5,86	14,93	0,73	31	80,00	36,26	3,73	0,22	10,20
anehwksu	90,53	1,60	17,33	0,77	31	86,26	12,80	14,66	0,23	13,40
autumnman	93,60	0,00	12,80	0,76	31	92,00	10,93	5,06	0,17	11,40
beaupowe	86,00	17,33	10,66	0,75	31	78,00	38,66	5,33	0,17	9,20
c,s,93/ksy	93,20	1,33	12,26	0,78	31	92,66	6,66	8,00	0,45	23,60
dhfpql,	94,40	0,00	11,20	0,76	31	95,73	1,33	7,20	0,20	11,20
dirdhfmw	96,93	0,00	6,13	0,75	31	98,13	0,80	2,93	0,18	11,20
difis wp	85,46	0,00	29,06	0,80	31	93,06	1,86	12,00	0,20	12,40
diltidgml	90,93	0,00	18,13	0,77	31	95,73	1,60	6,93	0,22	10,60
drizzle	92,13	6,66	9,06	0,77	31	87,46	21,06	4,00	0,20	11,60
dusru427	90,13	0,00	19,73	0,73	31	93,06	1,33	12,53	0,31	15,40
i love 3	94,93	1,06	9,06	0,78	31	91,06	10,66	7,20	0,14	8,60
love wid	88,80	14,13	8,26	0,75	31	84,40	27,20	4,00	0,20	11,80
loveis,	92,13	8,00	7,73	0,78	31	89,06	20,00	1,86	0,27	12,40
manseiii	83,06	18,40	15,46	0,74	31	74,00	46,13	5,86	0,20	13,00
rhkdwo	93,06	0,53	13,33	0,79	31	93,60	4,53	8,26	0,31	7,80
ria sua	97,20	1,86	3,73	0,76	31	89,73	16,80	3,73	0,18	10,80
tiddmswjd	90,93	0,26	17,86	0,76	31	91,20	2,40	15,20	0,26	14,40
tmdwnsl1	90,26	0,00	19,46	0,78	31	93,60	1,60	11,20	0,25	11,00
yuhwa1kk	97,06	0,00	5,86	0,76	31	97,33	0,00	5,33	0,17	11,80
min	52,80	0,00	3,73	0,72	31	55,73	0,00	1,86	0,14	7,80
Max	97,20	18,40	29,06	0,80	31	98,13	46,13	15,20	0,45	23,60
average	89,67	4,81	15,83	0,76	31	88,18	15,79	7,83	0,22	12,03

<Table 4> Ensemble 모델들의 결과

다는 것을 볼 수 있었다. 이는 최종 후보로 뽑힌 input vector들이 Ensem- YuFit-Select모델에서는 거리상으로 Ensem- UniFit-Fix모델보다 훨씬 가까이에 분포하고 있다는 것을 말하는 것으로 uniqueness 가 작다면, ensemble모델의 feature subset들이 각각의 classifier가 되었을 때, classifier들 간의 유사함을 가져오므로 바람직하지 않다.

6. Conclusion and Future work

6.1 Conclusion

결과를 요약한 표는 <Table6>와 같다. 진한 표시는 다섯 개의 모델 중에서 해당 데이터 셋에서 가장 좋은 accuracy를 보여준 모델을 나타내며, accuracy가 같은 경우 중복 표시 하였다. 위의 표에서 볼 수 있듯이 21개의 실험 데이터 셋 중 1개에서 Single 모델이 가장 우수한 accuracy를 보여주었다. 그러므로 classifier 하나만을 사용하여 만든 모델과 ensemble모델을 비교하여 볼 때 ensemble 모델이 accuracy를 높이면서 어느 데이터 셋에서나 기복없는 성능을 보여주었으므로 더 우수하다고 말할 수 있다. 그리고 ensemble 모델 두 개, Ensem-UniFit-Fix모델과 Ensem- YuFit-Select모델을 비교하여 볼 때, 두 모델의 accuracy 측면에서는 비슷한 결과를 보여주었다는 것을 알 수 있다. 하지만

Password	Models								
	Single-UniFit (Gamma = 0.2, u = 0.05)			Single-YuFit (Gamma = 0.2, u = 0.05)			Single-FF (Gamma = 0.2, u = 0.05)		
	Accuracy	FAR	FRR	Accuracy	FAR	FRR	Accuracy	FAR	FRR
90200idg	50.66	55.73	42.93	52.00	13.38	82.60	47.33	93.33	12.00
ahrfus88	85.60	13.33	15.46	55.53	1.10	87.81	80.00	40.00	0.00
anehwksu	85.73	23.20	5.33	73.54	2.61	50.29	81.33	37.33	0.00
autumnman	88.93	17.33	4.80	75.06	4.45	45.40	91.33	17.33	0.00
beaupowe	81.46	15.20	21.86	65.54	0.93	67.97	78.00	42.66	1.33
c,s,93/ksy	91.60	9.33	7.46	77.55	4.68	40.20	90.00	20.00	0.00
dhfpql.	94.13	11.46	0.26	95.44	6.90	2.21	90.66	18.66	0.00
dirdhfmw	96.53	6.93	0.00	87.09	0.79	25.01	90.66	18.66	0.00
difis wp	87.06	25.33	0.53	85.74	5.77	22.74	72.66	54.66	0.00
dltjdgml	91.46	16.53	0.53	81.53	7.08	29.85	87.33	25.33	0.00
drizzle	87.06	12.00	13.86	64.97	5.29	64.76	86.66	22.66	4.00
dusru427	88.40	23.20	0.00	86.62	1.88	24.87	80.66	38.66	0.00
i love 3	94.53	8.00	2.93	87.33	5.86	19.46	93.33	13.33	0.00
love wjd	79.46	28.26	12.80	72.55	6.19	48.70	76.66	44.00	2.66
loveis.	87.46	8.53	16.53	70.45	2.21	56.88	88.66	22.66	0.00
manseiii	72.80	11.46	42.93	56.59	2.13	84.67	84.00	29.33	2.66
rhkdwo	91.46	16.80	0.26	92.92	4.52	9.62	90.00	20.00	0.00
rla sua	95.20	7.46	2.13	55.29	2.35	87.04	86.00	28.00	0.00
tjddmswjd	90.40	18.93	0.26	86.20	4.81	22.77	80.00	40.00	0.00
tmdwnsl1	90.53	18.66	0.26	76.87	6.20	40.04	78.66	42.66	0.00
yuhwa1kk	96.13	7.73	0.00	95.57	1.64	7.20	92.66	14.66	0.00
min	50.66	6.93	0.00	52.00	0.79	2.21	47.33	13.33	0.00
Max	96.53	28.26	42.93	95.57	7.08	87.81	93.33	93.33	12.00
average	86.98	16.92	9.10	75.92	4.32	43.81	83.17	32.57	1.08

Ensem-

<Table 5> Single 모델들의 결과

	Ensem-UniFit-Fix	Ensem-YuFit-Select	Single-UniFit	Single-YuFit	Single-FF
90200idg	52.80	55.73	50.66	52.00	47.33
ahrfus88	89.60	80.00	85.60	55.53	80.00
anehwksu	90.53	86.26	85.73	73.54	81.33
autumnman	93.60	92.00	88.93	75.06	91.33
beaupowe	86.00	78.00	81.46	65.54	78.00
c,s,93/ksy	93.20	92.66	91.60	77.55	90.00
dhfpql.	94.40	95.73	94.13	95.44	90.66
dirdhfmw	96.93	98.13	96.53	87.09	90.66
difis wp	85.46	93.06	87.06	85.74	72.66
dltjdgml	90.93	95.73	91.46	81.53	87.33
drizzle	92.13	87.46	87.06	64.97	86.66
dusru427	90.13	93.06	88.40	86.62	80.66
i love 3	94.93	91.06	94.53	87.33	93.33
love wjd	88.80	84.40	79.46	72.55	76.66
loveis.	92.13	89.06	87.46	70.45	88.66
manseiii	83.06	74.00	72.80	56.59	84.00
rhkdwo	93.06	93.60	91.46	92.92	90.00
rla sua	97.20	89.73	95.20	55.29	86.00
tjddmswjd	90.93	91.20	90.40	86.20	80.00
tmdwnsl1	90.26	93.60	90.53	76.87	78.66
yuhwa1kk	97.06	97.33	96.13	95.57	92.66
average	89.67	88.18	86.98	75.92	83.17
Number of highest accuracy	10	10	0	0	1

<Table 6> 전체 모델의 accuracy와 number of the highest accuracy

YuFit-Select모델의 Classifier-Selection Method의 복잡성, ensemble member의 수의 일정하지 않음을 고려하여 볼 때 Ensem-UniFit-Fix모델이 알고리즘의 단순화를 가져와 좀 더 적용이 쉬운 모델이라고 말할 수 있다. Ensem-UniFit-Fix모델이 알고리즘을 복잡성을 줄이면서도 모델의 성능을 유지시킨 이유는 후보들 간의 degree of uniqueness로 설명할 수 있다. Ensem- UniFit-Fix모델은 후보들의 평균 degree of uniqueness가 다른 모델보다 훌륭하였으며, 이는 population의 diversity를 가져오는 역할을 하여 성능이 single모델보다 훌륭하였으며, Yu의 논문에서 제안한 모델과 비슷한 결과를 보여주었다.

6.2 Future work

제안한 FS-Ensemble의 문제점은 크게 두 가지로, fitness function에서 uniqueness 조건식과 accuracy 조건식을 단순히 더하는 방식을 취하였는데 이것을 multi-objective 관점에서 연구해 볼 필요가 있으며, training data를 전부 학습을 시키는 것이 아니라 본인의 key stroke pattern이라 할지라도 실수로 인한 outlier가 있는 상황이나 패턴을 입력 시에 사용자의 패턴이 바뀌는 상황을 대처하는 방법에 대한 연구가 필요할 것이다.

7. Reference

[1] 한대희 (1997), 암호 타자 패턴에 기반한 사용자 인증. 공학석사학위논문, 포항공과대학교
[2] E. Yu (2004). Constructing Response Model Using Ensemble Based On Feature Subset Selection. 공학박사학위논문, 서울대학교
[3] E. Yu, S. Cho (2004), Keystroke dynamics identity verification-its problems and practical solutions. Computers & Security, 23(5), pp428-440
[4] N. Srinivas, K. Deb (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation, 2(3), pp221-248, Fall
[5] D. Opitz (1999), Feature selection for ensembles. AAAI/IAAI, pp379-384.
[6] G. Brown, J. Wyatt, R. Harris, and X. Yao (2005), Diversity creation methods: a survey and categorisation. Information Fusion, 6(1), pp 5-20
[7] J. Yang, V. Honavar (1998), Feature Subset Selection using a Genetic Algorithm. in Feature Selection for Knowledge Discovery and Data Mining, H. Liu and H. Motoda (eds.), Kluwer Academic Publishers, pp117-136
[8] S. Cho, K. Cha (1996),Evolution of neural network training set through addition of virtual samples, International Conference on Evolutionary Computation, Nagoya, Japan, pp685-688
[9] S. Hashem (1997), Optimal Linear Combinations of Neural Networks. Neural Networks, 10(4), pp599-614
[10] R. M. Karp (1972), Reducibility among combinatorial problems, in R.E. Miller and J.W. Thatcher (eds.) Complexity of Computer Computations, Plenum Press, New York
[11] L. Breiman (1996), Bagging predictors. Machine Learning, 24(2), pp123-140.
[12] T. G. Dietterich (2000), Ensemble methods in machine learning. First International Workshop on Multiple Classifier Systems, pp1-15.
[13] Y. Freund, R.E. Schapire(1996), Experiments with a new boosting algorithm. Proceedings of the 13th International Conference on Machine Learning, Morgan Kaufmann, pp. 148156.
[14] J. Sullivan, J. Langford, R. Caruana, A. Blum (2000), Featureboost: A meta-learning algorithm that improves model robustness. Proceedings of the Seventeenth International Conference on Machine Learning.

[15] Y. Liu (1998), Negative correlation learning and evolutionary neural network ensembles. Ph.D. thesis, University College, The University of New South Wales, Australian Defence Force Academy, Canberra, Australia.
[16] X. Yao, Y. Liu (1998), Making use of population information in evolutionary artificial neural networks, in: IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 28, IEEE Press, pp. 417425.
[17] N. Sharkey, J. Neary, A. Sharkey (1995), Searching weight space for backpropagation solution types, Current Trends in Connectionism: Proceedings of the 1995 Swedish Conference on Connectionism, pp. 103120.