

이 재 원 교수지도
석사학위 청구논문

기계학습 기반의 주가시계열
예측에서의 성능 평가 연구사례

2005

성신여자대학교 교육대학원
교육학과 전자계산교육전공
문 정 희

기계학습 기반의 주가시계열
예측에서의 성능 평가 연구사례

이 재 원 교수지도

이 논문을 석사학위 논문으로 제출함.

2005년 5월

성신여자대학교 교육대학원

교육학과 전자계산교육전공

문 정 희

인 준 서

문 정 희 의 석사학위 논문을 인준함.

심사위원 (인)

심사위원 (인)

심사위원 (인)

성신여자대학교 교육대학원

논문 개요

주식시장의 비정상성(non-stationary)등으로 인해 주가시계열에서 가격변동이나 기대수익을 예측하는 것은 매우 어렵다. 주가를 예측하기 위한 일반적인 방법으로는 기업의 내재적 가치를 추정하는 기본분석과 과거의 주가패턴 등을 이용한 기술분석 기법 등이 사용되고 있다. 그러나 최근에는 보다 과학적이고 효율적인 방법으로 인공신경망(artificial neural networks)에 기반한 주가예측 연구가 활발히 진행되고 있다. 신경망을 이용한 주가예측은 지속적으로 이루어지고 있으나, 궁극적으로 투자자들에게 만족스러운 결과를 주고 있는지는 미지수이다.

이에 본 논문에서는 주가예측 연구사례들을 성능 평가의 측면에서 분석하고 이를 통해 기존 연구들의 통계적 신뢰도에 대해 검토하며 실제 투자에서의 활용 가능성에 대하여 논한다.

아울러 본 논문에서는 “NNShell”이라는 통합 시뮬레이션 도구를 통해 실제 거래상황과 밀접한 연관이 있는 주가예측에서 고려해야할 추가적인 요소와 다양한 성능 평가 척도들을 제시하고, 주가예측과 거래정책이 통합된 “통합 시뮬레이션”의 개념을 소개하며, “NNShell”의 개선을 위해 추가적으로 포함되는 것이 바람직할 것으로 판단되는 성능 평가 척도를 제안한다.

목 차

논문개요

I. 서론	1
II. 관련연구	3
1. 주가 예측에서의 일반적인 성능 평가 방법	3
1.1 시계열 접근법	3
1.2 거래 시뮬레이션 접근법	12
2. 주가시계열에 대한 예측 및 성능 평가 사례	15
3. 기존 성능 평가 사례의 요약 및 문제점.....	24
III. “NNShell”의 성능 평가 모형	27
1. 평가 모형의 개요	27
2. “주가예측 - 거래정책” 통합 시뮬레이션	32
3. “NNShell”을 이용한 성능 평가 분석 사례	34
IV. 추가적인 성능 평가 척도의 제안	47
V. 결론	62

참고문헌

ABSTRACT

표 목차

[표2-1] <사례연구 2>의 성능 평가 척도.....	18
[표2-2] <사례연구 3>의 성능 평가 방법.....	20
[표2-3] 시계열 접근법의 성능 평가 척도.....	24
[표2-4] 거래 시뮬레이션 접근법의 성능 평가 척도	25
[표3-1] 유동성 지표	28
[표3-2] 매수할인율에 따른 수익금의 변화.....	39
[표3-3] 이익실현율에 따른 수익금의 변화.....	40
[표3-4] 손절율에 따른 수익금의 변화.....	42
[표4-1] NNShell의 성능 평가 척도 및 거래정책 요소	47
[표4-2] NNShell의 시뮬레이션 결과파일 1.....	48
[표4-3] NNShell의 시뮬레이션 수행 결과 1.....	49
[표4-4] NNShell의 시뮬레이션 결과파일 2.....	54
[표4-5] NNShell의 시뮬레이션 수행 결과 2.....	55
[표4-6] NNShell의 시뮬레이션 결과파일 3.....	57
[표4-7] NNShell의 시뮬레이션 수행 결과 3.....	58
[표4-8] 월별 표준편차와 월별 적중률.....	61

그림 목차

[그림 3-1] 주가 차트	29
[그림 3-2] 지능형 거래 시스템의 일반적인 구조	32
[그림 3-3] 주가예측 - 거래정책의 통합 구조	32
[그림 3-4] NNShell의 시뮬레이션 수행 예제 1	34
[그림 3-5] NNShell의 시뮬레이션 수행 예제 2	35
[그림 3-6] NNShell의 시뮬레이션 수행 예제 3	36
[그림 3-7] NNShell의 시뮬레이션 수행 예제 4	38
[그림 3-8] NNShell의 시뮬레이션 수행 예제 5	40
[그림 3-9] NNShell의 시뮬레이션 수행 예제 6	42
[그림 3-10] NNShell의 시뮬레이션 수행 예제 7	43
[그림 3-11] NNShell의 시뮬레이션 수행 예제 8	44
[그림 3-12] NNShell의 시뮬레이션 수행 예제 9	45
[그림 4-1] <거래예제1>의 월별 수익률 그래프	50
[그림 4-2] <거래예제1>의 가설 그래프.....	51
[그림 4-3] <거래예제2>의 월별 수익률 그래프.....	56
[그림 4-4] <거래예제3>의 월별 수익률 그래프	59
[그림 4-5] <거래예제3>의 가설 그래프.....	60

I. 서론

주식시장의 비정상성(non-stationary), 잡음(noise), 비선형성(non-linearity) 등으로 인해 주가시계열에서 가격변동이나 기대수익을 예측하는 것은 매우 어렵고 복잡한 작업이다. 주가를 예측하기 위한 일반적인 방법으로는 기업의 내재적 가치를 추정하는 기본분석과 과거의 주가패턴 등을 이용한 기술분석 기법 등이 사용되고 있다. 기존의 방법들로 주가를 예측하고 효율적인 투자전략을 수립하는 것이 가능한지에 대해서는 아직까지 많은 논란이 제기되고 있다 [1].

최근에는 보다 과학적이고 효율적으로 주가를 예측하는데 인공신경망(artificial neural networks), 기계학습(machine learning)이론 등이 적용되고 있다. 신경망(neural networks)을 이용한 주가예측에 관한 연구는 국·내외에서 지속적으로 연구되고 있으나, 투자자들에게 만족스러운 결과를 주고 있는지는 여전히 미지수이다.

이에 본 논문에서는 연구사례의 분석을 통해 기존 주가 예측시스템의 성능 평가 척도들을 비교 분석하였다. 그 결과 대부분의 경우, 시스템의 성능을 평가하는 척도로 매우 단순한 척도들을 채택하고 있어 통계적 신뢰도에 대한 검토가 요구된다.

기존의 성능 평가 방법은 시계열 접근법과 거래 시뮬레이션 접근법으로 분류할 수 있는데, 거래 시뮬레이션 기법을 사용한 연구들이 보다 현실적인 요소들을 잘 반영하고 있는 것으로 판단된다. 본 논문에서는 “NNShell”이라는 통합 시뮬레이션 도구를 통해 실제 거래 상황과 밀접한 연관이 있는 주가예측에서 고려해야할 추가적인 요소와 다양한 성능 평가 척도를 제시하고, 주가예측과 거래정책이 통합된 “통합 시뮬레이션”의 개념을 소개한다.

실제 주식시장에서 투자자들은 자금의 한계로 투자에 제한이 따를 것으로 판단된다. 이에 본 논문에서는 최대 월별 누적 수익률을 400%로 가설을 설정하여 주가 예측시스템의 성능을 분석하였다. 그 결과 “NNShell”에 포함되는 것이 바람직할 것으로 판단된 추가적인 성능 평가 척도를 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 주가예측 연구사례들을 성능평가의 관점에서 비교 분석하고, 제 3장에서는 실제 주식시장의 현실적인 측면을 고려한 “NNShell”의 성능 평가 모형을 소개하고, “NNShell”을 통해 통합 시뮬레이션의 개념을 소개함과 아울러 성능 평가 척도들의 최적화 과정을 실제 데이터를 적용한 실험결과를 통해 예시한다. 제 4장에서는 “NNShell”에 포함되는 것이 바람직할 것으로 판단되는 추가적인 성능 평가 척도를 제안한다. 마지막으로 제 5장에서 본 논문의 결론을 제시한다.

II. 관련 연구

1. 주가 예측에서의 일반적인 성능 평가 방법

주가 예측시스템의 개발과정에서는 두 단계에서 예측 알고리즘의 성능을 측정할 수 있는 척도를 필요로 한다. 첫 번째 단계로, 예측 알고리즘의 모델(model)을 구축하는 단계에서는 모델에 관련된 파라미터들을 최적화하기 위해 성능 측정 척도가 필요하고, 두 번째 단계로, 최종 알고리즘을 테스트하는 단계에서 테스트(test) 데이터에 대해 알고리즘을 적용하여 알고리즘의 최종 성능을 검증하기 위해 성능 측정 척도(metrics)가 필요하다. 본 논문에서는 최종 성능 검증 단계에서의 성능 측정 척도를 중심으로 사례 분석을 수행할 것이므로 이들 척도에 대하여 살펴보기로 한다. 성능 검증 단계에서의 평가 방식은 크게 시계열 접근법과 거래 시뮬레이션 접근법의 두 가지로 구분된다.

1.1 시계열 접근법

시계열이란 어떤 관측치 x 가 시간(t)의 흐름에 따라 변화하는 것을 말한다. 즉, 주가의 시계열 분석이란 주가의 이동을 시간의 흐름에 따라 분석하는 것을 말한다. 주가 예측 문제를 고전적인 방식으로 정의하자면, 주가의 종가(close)들로 구성된 시계열 $y(t)$ 를 구성하는 것이다. 이로부터 수익을 측정하는 가장 일반적인 방식은 아래와 같이 정의 되는 h 일간의 수익 $R_h(t)$ 를 사용하는 것이다[2].

<정의1> 주식의 일반적인 수익 $Rh(t)$

$$Rh(t) = \frac{Close(t) - Close(t-h)}{Close(t-h)}$$

본 논문에서는 시계열 Rh 를 y 로 표기하기로 한다. 또한 y 는 k 개의 이전 값들에 대한 어떤 함수 g 인 것으로 가정한다. 미래에 h 일간의 수익을 예측하는 것은 다음과 같다.

<정의2> h 일간의 수익

$$y(t+h) = g(y(t), y(t-1), \dots, y(t-k))$$

모델링 과정은 측정된 데이터의 최고 근사치 함수 g 를 찾는 것이다. 즉, g 를 찾는 것은 파라미터 최적화의 과정이다. 알려지지 않은 함수 g 는 다양한 방법으로 찾을 수 있다. 예를 들면 자귀회귀공식(AR), 인공 신경망(artificial neural network) 등이 있다. 알려지지 않은 파라미터들은 일반적으로 RMSE(root mean squared error)에 기반 하여 계산된다. RMSE의 정의는 아래 <정의3>과 같다.

<정의3> $RMSE$

$$RMSE = \sqrt{\frac{1}{N} \sum (g(t) - y(t+h))^2}$$

결국, RMSE를 최소화하는 것이 주가예측 작업의 최종 목표가 된다. 주가예측을 활용하기 위해서는 정해진 규칙에 의거하여 거래 행위를 결정하게 된다. 일반적인 규칙은 아래와 같다.

<정의4> 예측 결과를 활용하기 위한 간단한 규칙	
$D(t) =$	$\begin{cases} Buy & : \text{if } g(t) > \alpha \\ Sell & : \text{if } g(t) < -\beta \\ Do\ nothing & : \text{if } g(t) = 0 \end{cases}$

α 와 β 는 예측된 주가 변화에 따라 매수, 매도에 대한 임계치 파라미터이다. 시계열 접근법에 의한 예측은 일반적으로 고정된 시간 간격(fixed horizon)에 의해 평가 되어지지만, <정의4>의 거래 규칙에 적용된 임계치들은 변경되어 사용될 수 있다. 고정된 시간 간격에서 알고리즘의 성능을 측정하기 위한 척도는 예상된 값들과 실제 h일 후의 결과를 비교하는 것이다. 시계열 접근법에서의 주가의 예측은 시간(t)에 의해 아래와 같이 표현될 수 있다.

예측된 가격	실제 가격
$\{\hat{C}lose(t), t = h + 1, \dots, N\}$	$\{Close(t), t = 1, \dots, N\}$

h 일간의 예측된 수익(return)과 실제 수익은 아래 표와 같다.

h일간의 예측된 수익	h일간의 실제 수익
$\{\hat{R}(t), t = h + 1, \dots, N\}$	$\{R(t), t = h + 1, \dots, N\}$

예측된 가격($\hat{C}lose(t)$)과 예측된 수익($\hat{R}(t)$)은 (t-h)에서의 종가 $Close(t-h)$ 와 의 비교에 의해 산출된 것이다.

위와 같은 정의 하에, 시계열 접근법에서의 예측에 대한 성능을 평가하는 주요 척도를 제시하자면 다음과 같다.

1) RMSE

좋은 예측 알고리즘은 RMSE가 가능한 한 낮아야한다. 즉, RMSE가 낮으면 낮을수록 좋은 예측 알고리즘이다.

<정의5> RMSE
$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (Close(t) - \hat{Close}(t))^2}$

2) 적중률 (Hit Rate H_R)

예측기(predictor)의 적중률은 수익이 얼마나 정확하게 예측되었는지를 표현한다. 0이 아닌 예측들 중에 적중한 수와 주가 시계열에서 0이 아닌 변동을 나타낸 총수와의 비율이다.

<정의6> 적중률 (Hit Rate H_R)
$\frac{\left \{R(t)\hat{R}(t) > 0\}_{h+1}^N \right }{\left \{R(t)\hat{R} \neq 0\}_{h+1}^N \right }$

적중률 계산에서 0인 예측과 0인 수익(returns)을 제외시키는 이유는 다음과 같다. 만일 0인 예측과 수익을 포함한다면 아래와 같은 5개의 조합들도 적중률 계산에 포함되어야 한다.

0을 포함하는 5가지 조합	
$\hat{R}(t) = 0$	$R(t) > 0$
$\hat{R}(t) = 0$	$R(t) < 0$
$\hat{R}(t) = 0$	$R(t) = 0$
$\hat{R}(t) > 0$	$R(t) = 0$
$\hat{R}(t) < 0$	$R(t) = 0$

이러한 조합들을 적중률 계산에 포함할 경우, 경우에 따라 양(positive)의 결과와 음(negative)의 결과의 수가 비대칭의 형태로 나타날 수 있다. 대개의 주가 시계열의 경우, 전체 데이터의 20% 정도가 0의 변동을 보인다. 이들 데이터가 “up fraction”에 포함될지 혹은 “down fraction”에 포함될지에 따라 적중률에 무의미한 영향을 미치게 되는 것이다. 위 조합에 해당하는 데이터를 모두 제거하면, “up fraction”은 50%에 매우 가까워진다. 그러므로 하루 수익의 경우 즉, h=1인 경우, 적중률 H_R 이 0.5보다 커질수록, 예측력이 매우 좋은 것으로 간주될 수 있으며 높은 수익을 기대할 수 있다. 위에 언급한 일반적 H_R 은 두 가지로 확장하여 적용할 수도 있다. H_{R+} 과 H_{R-} 는 아래 <정의7>과 같다.

<정의7> Hit Rate H_R 의 확장	
$H_{R+} = \frac{ \{R(t) > 0 \text{ AND } \hat{R}(t) > 0\}_{h+1}^N }{ \{\hat{R}(t) > 0\}_{h+1}^N }$	$H_{R-} = \frac{ \{R(t) < 0 \text{ AND } \hat{R}(t) < 0\}_{h+1}^N }{ \{\hat{R}(t) < 0\}_{h+1}^N }$

이 확장에서는, 수익 R을 주가가 상승한 경우(+)와 하락한 경우(-)로 구별하여 예측 결과를 측정하고 있으며 이는 매우 유의미한 시도로 보인다. H_{R+} 과 H_{R-} 중 어떤 것이 투자자들에게 유익한 정보일까? 투자자들의 주식 투자

목적은 이익(profit)을 창출하는 것이다. 어떤 주식이 오를 것이라고 예상한다면, 투자자들은 그 종목에 투자하여 이익을 창출할 것을 기대한다. 그러므로 투자자들의 직접적인 행동에 영향을 미치는 것은 H_{R+} 일 것이다.

3) 순이익 (Net Profit)

예측 알고리즘의 궁극적인 성공 여부는 실제 거래에 적용하여 이익(profit)을 창출할 수 있는가의 여부에 달려있다. 순 이익(P_n)은 여러 예측에 대한 결과를 종합하여 표현한 것으로 각 거래의 결과를 합산하는 것이다. 이는 <정의 8>에 의해 계산할 수 있으며, 이 경우 가격이 하락하는 것을 맞춘 경우도 수익을 얻을 것으로 간주함을 알 수 있다.

<p style="text-align: center;"><정의8> 순 이익 (Net Profit)</p> $P_n = \sum_{t=h+1}^N (Close(t) - Close(t-h)) \cdot sign(\hat{Close}(t) - Close(t-h))$

4) 단순 주가 예측 (Naive Prediction of Stock Prices)

단순 예측기(naive predictor)는 단순히 미래 가격 $\hat{close}(t+h)$ 를 오늘의 가격 $close(t)$ 이라고 예측한다. 즉, 오늘의 주가가 10,000원이라면 단순히 미래의 가격도 10,000이라고 예측하는 것이다. 이 예측기는 그 자체로는 의미가 없지만, 주어진 예측기와의 상대적인 비교를 통해 예측기의 성능을 평가할 수 있는 대상이 될 수는 있다. 예측기의 성능을 측정하는 주어진 예측기가 단순 예측기와 얼마나 다른지를 측정하는 계수인 Theil 계수(coefficient), T_c 에 의해 행해지며, 성능을 평가하고자하는 예측기에 의해 예상된 가격의 RMSE와 단순 예측기의 RMSE 사이의 몫으로 정의된다. Theil 계수는 ‘정보 계수

(information coefficient)' 혹은 't-test'라고도 불린다.

<정의9> T_c (Theil Coefficient of inequality of T_c)
$T_c = \frac{RMSE_y}{RMSE_{yN}} = \frac{\sqrt{\sum_{t=h+1}^N (Close(t) - \hat{Close}(t))^2}}{\sqrt{\sum_{t=h+1}^N (Close(t) - Close(t-h))^2}}$

<정의9>에 의하면, 주어진 예측기와 단순 예측기의 RMSE를 비교하여, $T_c > 1$ 인 경우 주어진 예측기가 단순 예측기에 비해 성능이 좋지 않다는 것을 의미하며, 반면에 $T_c < 1$ 인 경우 그 예측기는 단순 예측기에 비해 성능이 더 좋다고 말할 수 있다.

5) ϵ -증가 예측기 (ϵ -increase predictor)

가격을 예측하는 단순 예측기와 같은 취지로 주가의 등락, 즉 수익을 예측하는 것이 ϵ -증가 예측기이다. 이 예측기는 미래의 가격 $close(t+h)$ 를 오늘 가격에 매우 작은 양의 수인 ϵ 을 더한 $close(t)+\epsilon$ 로 예측한다. 결국 미래의 수익이 ϵ 이라고 보는 것이다. 따라서 이 예측기는 무조건 주가가 아주 조금 오른다고 예측하는 셈이 된다. 이 예측기의 적중률 H_ϵ 은 <정의10>과 같다.

<정의10> ϵ -increase
$H_\epsilon = \frac{ \{R(h) > 0\}_{h+1}^N }{ \{R(h) \neq 0\}_{h+1}^N }$

이 공식에 의하자면, ϵ -증가 예측기의 적중률은 단순히 실제 결과가 양의 수익을 보인 경우의 비율이 된다. 어떤 예측기의 적중률을 이 예측기와의 상대 비교를 통해 평가하는 공식은 <정의11>과 같다.

<정의11> HR_ϵ 증가 예측기의 적중률 비교

$$HR_\epsilon = \frac{H_R}{H_\epsilon}$$

<정의11>에 의해, 주어진 예측기와 ϵ -증가 예측기의 적중률을 비교하면, $HR_\epsilon < 1$ 인 경우 주어진 예측기가 ϵ -증가 예측기에 비해 성능이 좋지 않다는 것을 의미하며, 반면에 $HR_\epsilon > 1$ 인 경우 그 예측기는 ϵ -증가 예측기에 비해 성능이 더 좋다고 말할 수 있다.

6) 단순 수익 예측기 (Naive Prediction of Returns)

Theil 계수의 정의인 <정의9>는 단순 예측기와의 비교를 통해 주어진 예측기의 성능을 평가한다. 단순 수익 예측기의 개념도 유사하다. 단순 수익 예측기는 오늘의 수익 ($R_h(t)$) (시간 단계 $t-h$ 이후 현재의 수익)으로 미래 수익 $R_h(t+h)$ 로 예측하는 것이다. 단순 수익 예측기의 RMSE와 주어진 수익 예측기의 RMSE 비율을 측정하는 척도로 유의미한 척도이며 그 정의는 아래와 같다.

<정의12> T_r (Theil Coefficient of inequality for returns T_r)

$$T_r = \frac{RMSE_r}{RMSE_{rN}} = \frac{\sqrt{\sum_{t=h+1}^N (Rh(t) - \hat{R}h(t))^2}}{\sqrt{\sum_{t=h+1}^N (Rh(t) - Rh(t-h))^2}}$$

<정의12>에 의하자면, 주어진 예측기와 단순 수익 예측기의 RMSE를 비교하면, $T_r > 1$ 인 경우, 주어진 예측기는 단순 수익 예측기에 비해 성능이 좋지 않

다는 것을 의미하며, 반면에 $T_r < 1$ 인 경우 그 예측기는 단순 수익 예측기에 비해 성능이 좋은 것으로 평가될 수 있다.

단순 수익 예측기의 적중률 H_N 은 다음과 같다.

<p style="text-align: center;"><정의13> H_N (단순 수익 예측기의 적중률)</p>
$H_N = \frac{ \{R_N(t)R_N(t-h) > 0\}_{t=1}^N }{ \{R_N(t)R_N(t-h) \neq 0\}_{t=1}^N }$

단순 수익 예측기는 현재 수익이 나고 있으면 이후에도 주가가 상승할 것으로 판단하고, 현재 수익이 음의 값이면 이후에도 주가가 하락할 것으로 판단하는 것이다.

HR_N 은 주어진 예측기의 적중률과 단순 수익 예측기의 적중률의 비율로 정의된다. 따라서 HR_N 은 예측기의 적중률과 단순 수익 예측기의 관계를 비교하는 척도이다.

<p style="text-align: center;"><정의14> HR_N</p>
$HR_N = \frac{H_R}{H_N}$

위 공식에 의해, 주어진 예측기와 단순 수익 예측기의 적중률을 비교하면, $HR_N < 1$ 인 경우, 주어진 예측기는 단순 수익 예측기에 비해 성능이 좋지 않은 것을 의미하며, 반면에 $HR_N > 1$ 인 경우, 그 예측기는 단순 수익 예측기에 비해 성능이 더 좋다고 말할 수 있다.

7) Buy-and-hold

Buy-and-hold에서 이익 P_b 는 기간 $\{1, \dots, N\}$ 과 개별 종목(one particular

stock)에 대해 아래와 같이 정의된다.

<정의15> Buy-and-Hold의 이익(Profit) P_b
$P_b = Close(N) - Close(1),$

즉, 매수했을 때와 매도했을 때의 종가의 차이를 이익으로 간주한다. P_r 은 아래 <정의16>과 같이 예측기의 순이익(P_n)과 buy-and-hold의 이익(P_b)간 비율로 정의된다.

<정의16> 순 이익과 buy-and-hold의 수익 P_b 의 비교
$P_r = \frac{P_n}{P_b}$

<정의16>에 의해, 예측기의 순이익(P_n)과 Buy-and-hold의 P_b 를 비교하면 $P_r < 1$ 인 경우, 예측기의 순 이익은 “Buy-and-Hold”에 의한 이익에 비해 낮은 것을 의미한다. 반면에 $P_r > 1$ 인 경우 예측기는 더 높은 순이익을 나타낸다.

이러한 비교에 의해 어떤 예측기의 예측에 의한 수익이 진정한 예측력에 의한 수익인지 아니면 단순히 일반적인 시장의 추세 때문에 얻어진 수익인지를 테스트할 수 있다.

1.2 거래 시뮬레이션 접근법

거래 시뮬레이션(trading simulation) 접근법은 실제 거래와 최대한 유사하게 구현된 시스템을 통해 거래 규칙을 기초로 한 예측 알고리즘으로 구현하는 방법이다. 거래규칙은 아래 <정의17>과 같다[3].

<정의17> 거래 규칙

$$T(t) = \begin{cases} Buy & : \text{if } g(t) > 1 \\ Sell & : \text{if } g(t) < -1 \\ Do\ nothing & : \text{if } g(t) = 0 \end{cases}$$

거래 규칙은 매수, 매도에 대한 기준을 의미한다. 거래 시뮬레이션 접근법은 앞서 제시한 시계열 접근법보다 실제적인 상황을 고려함으로써 더 높은 목표를 성취할 수 있다. 거래 시뮬레이션 접근법에서는 거래세를 포함하여 결과를 산출하며, 복수의 거래는 하나씩 차례대로 이루어질 수도, 동시에 이루어질 수도 있다. 거래시스템의 결과는 비교대상에 대한 증가 수치와 연간이익으로 나타낸다. 전체 주식 목록에 대한 비교를 도표로 나타내고, 결과 값의 비교를 통해 시스템의 성능을 측정할 수 있다. 거래 시뮬레이션 접근법에서 성능 척도들은 다음과 같이 나타낼 수 있다.

1) 전체 이익 (Total Profit)

전체 이익(total profit)은 거래자가 직관적인 느낌과 거래지표를 가지고 데이터 학습기간의 시뮬레이션 거래를 통해 누적된 총수익을 말한다.

2) 일별 평균 이익 (Average Profit per Day)

전체 이익 측정에서 축적의 효과를 얻기 위해 일별 평균 이익을 계산하여 거래 성능을 평가하는데 사용한다.

3) 연간 이익 (Profit per Year)

거래 알고리즘을 적용해서 성취된 연간 이익을 계산한다. 연간 이익은 전체

기간동안 측정되어 사용된다. 연간 이익은 매년의 성능을 파악할 수 있다.

4) (수익성이 있는 거래들) Fraction Profitable Trades

시뮬레이션 결과의 평가 기준을 찾기가 어렵지만, 실제 거래에서 얼마나 좋은 성능을 나타낼 수 있을지 예상할 수는 있다. 또한 실제로 일어나는 거래 결과가 발생하는 위치에 대한 정보를 제공해주기도 한다. 만약 시뮬레이션을 통해 높은 수익을 얻었다라면, 이것은 운 좋게 높은 수익을 산출한 몇몇의 거래들 때문인지, 전체의 고른 거래들 때문인지 의심해 볼 필요가 있다. 따라서 수익성 있는 거래들에 대한 수익률을 1년 주기로 계산하는 것은 성능 평가 있어서 매우 중요한 부분이다[4].

5) 거래 수 (Number of Trades)

시뮬레이션이 진행되는 동안 생성된 거래 횟수가 작다면, 거래 규칙 기반 시스템으로부터의 추측된 이익의 통계적인 의미는 미약하다. 그러므로 거래 횟수는 매우 중요하다. 그리고 거래 결과와 함께 병행 기록되어야 할 필요가 있다.

주가 예측시스템의 성능 평가 방법은 이와 같이 시계열 접근방법과 거래 시뮬레이션 접근방법으로 크게 두 가지로 분류될 수 있다. 다음 2절에서 주가 예측에 대한 사례연구를 통해 성능 평가의 방법을 구체적으로 분석한다.

2. 주가시계열에 대한 예측 및 성능 평가 사례

<사례연구1>

연구 [5]에 의하면 인공 신경망(artificial neural network)은 주가예측에 효과적으로 활용될 수 있다.

각각의 주식에 대해 분리된 네트워크를 구축하는 방법과 모든 주식에 대해 같은 네트워크를 공유하는 두 가지 방법 모두를 실험해 보고 각각의 성능을 비교하였다. layer(층)를 공유하는 경우와 그렇지 않은 경우의 비교 실험도 수행하였다. 부분적인 파라미터들을 교차하여 주가 데이터를 공유하는 학습 방식을 이 연구에서는 ‘multi-task learning’ 형식이라고 부른다[5].

토론토 주식 시장(Toronto Stock Exchange)에서의 실험을 위해 35개의 위험 자산과 1개의 안정적 자산, 8년 동안의 학습 자료(1986년 2월-1994년 1월)와 5년 동안(1989년 2월-1994년 1월)의 테스트 자료가 사용되었다. 인공 신경망의 학습을 통해 주식의 미래결과를 예측하는 것의 최종 목표는 예측 에러를 최소화하는 것이다. 여기서 학습은 궁극적으로 이러한 예측 에러를 최소화하는 방향으로 목표 함수 $f(x)$ 를 갱신해 나가는 것이다. ‘multi-task learning’의 목표는 데이터의 학습 속도를 향상시키는 것이 아니라 일반적인 성능을 향상시키는 것이다. 이 연구에서는 금융 데이터의 비정상성(non-stationarity)과 경제 시계열의 특성들을 고려하였고 다양한 경제상황에서의 성능을 평가하였으며, 거시 경제 변수¹⁾와 미시 경제 변수²⁾ 모두를 고려하였다.

1) 거시경제변수란 경제 전반의 상황을 알려주는 것으로 국내총생산, 물가, 실업율, 금리, 환율, 국제수지 등을 말한다.

2) 개인의 소비나 저축과 같은 경제행위를 분석하고 기업의 경제활동을 분석하는 것을 말한다.

학습 알고리즘은 신경망 파라미터 최적화에 기초하였다. 신경망의 결과(output)를 기반으로 하여 거래 모듈(trading module)의 최적화도 수행하였다. 2개의 다른 신경망구조를 가지고 4개의 다른 형태의 파라미터를 공유하는 실험을 수행하였다. 여기서 4가지 형태란 모두 공유(all sharing), 첫 번째 은닉층 공유(1st hidden layer), 출력층 공유(output layer), 공유하지 않음(not sharing)을 의미한다. 위험분석 요인으로는 '베타(beta)'값이라는 수치를 사용하여 시장의 위험성을 수치로 표현하였고, '알파(alpha)'값이라는 수치를 사용하여 위험분석(risk analysis)을 고려하였다. 'TSE300 지수'와 이 연구의 결과를 활용한 'buy-and-hold'의 두 가지 벤치마크를 비교한 결과에 의하면, 각각 연간 4.4%와 8.3% 정도로 'buy-and-hold'의 성능이 더 우수하였다. 이 결과는 본 논문에서 앞서 언급한 시뮬레이션 기법에 의한 성능 평가로 보인다. 위 실험 결과에 의하면, 부분적인 파라미터의 공유를 허용한 경우가 가장 성능이 좋은 것으로 나타났다. 'buy-and-hold' 또는 'TSE300 지수'와 매년의 위험 결과를 비교하였을 때, 최고의 벤치마크보다 14%정도 높은 결과를 나타냈다. 부분적인 파라미터의 공유와 함께 거시 경제 변수, 미시 경제 변수에 대한 고려도 효과적인 학습에 긍정적인 영향을 미친 것으로 나타났다.

결과적으로, <사례연구1>은 앞서 언급한 시뮬레이션 기반의 성능 평가 방법을 사용한 것으로 판단되며, 두 개의 벤치마크(benchmark)를 비교하여 성능 평가를 수행하였다. 위험분석 요인을 성능 평가 방법에 고려한점은 유의미한 시도로 판단되나, 단지 연간 수익률만을 가지고 시스템의 성능을 평가하는 매우 단순한 평가 척도를 채택하고 있다.

<사례연구2>

연구 [6]에서는 다중 인공신경망(multi neural network)의 역전파 알고리즘(back propagation)을 사용하여 재학습을 수행하고 함수의 기본적인 변화를 설명하였다. 또한, 전처리(preprocessing)와 후처리(postprocessing)과정에서 기술분석(technical analysis)이 전체 시스템 성능 향상에 영향을 미치는 지의 여부를 실험하였다. 이 연구에서는 최종적으로 S&P 500 index에 대한 거래를 수행하는 거래시스템을 제안하였다.

연구의 목적은 만일 시스템에 부가적인 요소를 포함하는 것이 더 정교한 데이터 필터링(data filtering)과 부분적인 결과들의 통합에 의해 더 향상된 결과를 나타내는지 실험하였다. 이 연구에서는 시스템의 데이터를 구분하기 위해 잘 알려진 기술 지표(technical indicator)를 사용하였고, 이익이 없는 거래 횟수를 감소시키기 위해 추가적인 지표(additional indicator)를 사용하였다. 여러 개의 시스템을 결합하여 실험하고 5년이 넘는 거래 기간에 걸쳐 테스트를 수행하였다.

데이터 처리의 필터링(filtering) 방법으로는 'ADX-based'라는 필터를 사용하였다. 출력된 예측치를 분석하여 '매수'와 '매도'를 추천하였으며, '규칙-기반(rule-based)'의 통합 전략을 이용하여 단계별로 실험을 수행하였다.

주식시장에서의 성능 측정의 표준은 예측 모델이 얼마나 많은 수익을 얻을 수 있는지를 측정하는 것이다. <사례연구2>에서는 아래 표와 같이 성능 측정 방법으로 'ARR'과 'BETC'를 제시하였다.

성능 측정 (Performance Measures)	
① ARR (Annual rate of return)	$ARR = \frac{k}{n} \sum_{i=1}^n ri$
② BETC (Break-even-transaction cost)	$BETC = \frac{1}{m} \sum_{i=1}^n ri$
비 고	<ul style="list-style-type: none"> · n: 총 거래시간, k: 1년간의 거래시간 · ri: 시간단위에서 i에서 수익률 m: 총 거래 횟수

[표 2-1 <사례연구2>의 성능 평가 척도]

예측시스템의 성능은 'ARR'에 의해 직관적으로 판단할 수 있다. 그러나 'ARR'이 같은 20%를 달성하였다고 하더라도 100번의 거래로 20%를 달성한 것과 50번의 거래로 20%를 달성한 것은 거래당 수익률에서 큰 차이를 보이기 때문에 거래당 수익률을 고려해야 한다. 연구 [6]에서는 거래당 수익률인 'BETC'를 제시하고 있다. 'BETC'를 측정하여 수익이 없는 거래 횟수를 줄여 나감으로써 거래당 수익률을 높였다.

연구 [6]에서는 'ADX-based' 방향의 패턴필터를 사용하는 것과 '규칙-기반(rule-based)의 통합전략'을 가지고 단계별로 실험을 수행하였다. 기술분석에 의해 $ADX^1(18,18,15)$, $ADX^1(13,13,10)$, $ADX^2(3,3,65)$ 의 세 가지의 경우를 사용하도록 추천되었고, 이 중 가장 좋은 성과를 나타낸 것은 $ADX^2(3,3,65)$ 로 ARR이 12.4%를 나타내었다. 그러나 BETC는 0.14%에 불과하였으므로 ARR에 비해 상대적으로 거래당 수익률이 낮게 나타났다. 가장 좋은 성과를 나타낸 $ADX_2(3, 3, 65)$ 로 통합실험을 진행하였다. 시스템을 6가지형태로 나누어 실험을 진행하였다. 시스템 A는 '규칙기반', B는 '투표전략', C는 'NN전략', D는 'MACD 확인 전략', E는 'MACD'만을 사용하고, 마지막으로 F는 '임계치 필터 규칙통합'을 사용하여 실험을 수행하였다. 실험결과는 A에서 ARR

12.4%로 비교적 높은 성취를 나타내었지만 432번의 많은 거래횟수로 거래당 수익률은 상대적으로 낮았다. *D*는 ARR이 15.99%, BETC는 1.49%로 최상의 결과를 나타내었다. *D*는 'MACD 필터'로 여겨지며, 이익이 되지 않는 거래 수를 감소시켰다. *D*와 *C*를 비교해보면 ARR의 상당한 개선뿐만 아니라 거래 횟수를 152번에서 52번으로 감소시켰다. 또한, F(threshold filter/rule integration)는 프로세싱 단계에서 기술적인 지표를 사용하지 않았지만, 성취된 ARR은 13.35%, BETC는 0.53%로 상당히 개선된 결과를 나타내었다. 위 실험을 통해 나타난 결과를 종합하면 다음과 같다. 패턴 필터링을 과정에 'ADX' 지표를 사용하였을 때, 좀더 작은 파라미터 값을 사용한 것이 성능이 개선되긴 하였지만, 임계치 기반(threshold-based)의 성능에는 미치지 못하였다. 이것은 'ADX'가 시장의 급격한 변화에 신속히 반응하지 못하기 때문으로 판단된다. 'MACD' 지표를 사용하여 이익이 없는 거래 횟수를 감소시키는 후처리 과정을 통해서 높은 이익률을 실현할 수 있었다. 필터링 단계에서 잘 알려진 기술적인 지표를 사용하고 이익이 없는 거래 횟수를 줄이기 위해 추가적인 지표를 사용하는 시스템이 최상의 결과를 나타내었다. 성능이 가장 좋은 시스템은 54번의 거래로 15.99%의 ARR(annual rate of return)을 나타냈다. 신경망 기반의 증권거래 시스템에 주가의 가격 및 거래량의 정보를 이용하는 기술적인 분석을 적용하였고 이 방법은 시스템의 예측 성능을 향상시켰다.

결과적으로, <사례연구2>는 본 논문의 앞에 언급한 시뮬레이션 기반으로 판단되며, 성능 평가 척도들로는 'ARR'로 연간 수익률을 나타내었고, 거래 횟수를 고려한 거래당 수익률인 'BETC'를 제시하였다. <사례연구2>는 <사례연구1>에 비해 거래당 수익률을 고려함으로써 개선된 성능 평가 척도를 사용한 것으로 판단된다.

<사례연구3>

연구 [7]에 의하면 웹 게시판의 원문 자료를 사용하여 금융 시장의 예측을 두 가지 방법으로 접근하였다. 첫 번째는 ‘최대 엔트로피 텍스트 분류 (maximum entropy text classification)’를 기초로 하여 예측하고, 두 번째는 거래량, 게시된 메시지의 수, 그리고 하루에 게시된 단어의 총수 등의 수치 데이터만을 사용하여 간단한 규칙(rule)들을 학습하기 위해 유전자 알고리즘 (genetic algorithm)을 사용하였다.

두개의 예측기를 통합하는 것이 보다 더 우수한 결과를 나타내었다. 예측기의 성능을 개선시키는 ‘GA 방법’을 시도하였다. 금융 데이터에 계산적인 지능을 적용하는 고전적인 방법에는 과거의 가격 데이터를 기반으로 미래의 결과를 예측하는 방법이 있다. 그러나 최근에는 자료의 새로운 소스를 활용할 수 있게 되었는데 금융상의 많은 텍스트 자료뿐만 아니라 웹상의 자료도 활용할 수 있게 되었다. 이 연구는 수치 데이터에 기반하는 간단한 규칙을 학습하기 위해 유전자 알고리즘을 사용하는 방법과 텍스트를 분석하는 방법을 통합하는 것을 목표로 실험을 진행하였다.

	최대 엔트로피 텍스트 분류 (A)	유전자 알고리즘 (B)	통합 (C)
모든 종목(22)	-8.11%	-6.22%	2.88%
10k가 넘는 게시물 (12 stocks)	6.91%	5.948%	19.26%

[표 2-2 <사례연구3>의 성능 평가 방법]

(A)의 최대 엔트로피 텍스트 분류방법은 전체 언어자료의 학습을 통하여 각

문서에 대해 일치할 단어수를 계산하여 큰 행렬을 만드는 것이 각각의 차수를 특정한 단어와 일치하는 N개 차수 (n-dimensional)의 공간으로서 나타낼 수 있다. 200일이 넘는 거래일 동안 발생한 수익률의 평균을 전체 22개의 주식과 10k가 넘는 12개의 주식으로 분류하여 결과를 나타내었다. 전체 종목에 대해서는 주가가 하락하였으나 10k가 넘는 게시물을 갖는 12개의 주식에 대해서는 약간의 상승세로 6.91%를 나타내었다. 이 결과는 가능성이 있는 것으로 판단되나, 아직까지 기대에 미치지 못하였다. 최대 엔트로피 텍스트 분류는 알고리즘의 문서에 대한 “yes/no”로 단순한 대답만을 제공하지는 않는다. 고전적인 문서들의 분류에 따르면 0.5보다 큰 값들은 상승할 가능성이 있다고 예측한다. 거래전략에 의해 임계치(threshold)구간을 생성하였는데, 실행결과를 살펴보면 대략 0.4와 0.5 사이의 임계치를 사용하며 10k가 넘는 게시물의 주식의 경우 20%와 30% 사이에서 주가가 상승하였다. 그러나 전체 주식에 대해서는 5%가 넘는 하락세를 나타냈으므로 개선이 필요하다.

(B)의 유전자알고리즘 방법은 원문의 자료 값을 찾는 것에 의미를 두어, 규칙학습자를 위해 과거의 가격을 입력 자료로 사용하지 않고 그 대신에 거래량을 사용한다. 메시지 수와 전체 단어 수는 하루 동안에 게시된 수를 나타내며 과거의 정보 도표로부터 ‘매수/보유’ 그리고 ‘매도’의 신호(signal)를 다음의 형식에 의해 규칙적으로 나타냈다. 만약 $v > k * ma(v, n)$ 일때, 여기서 v는 오늘의 거래량이고, $ma(v, n)$ 은 n거래일에 산출한 과거 거래량 값들의 평균이고, k는 임의 상수이다. 이 규칙에 따라 오늘의 거래량보다 이전 거래량 평균에 상수를 곱한 값이 작으면 ‘매수/보유’ 신호를 보내고, 반대의 경우 ‘매도’ 신호를 보낸다. 유전자 알고리즘의 거래량을 사용한 (B)의 결과는 (A)의 결과와 유사하였다. 22개의 전체 목록에 대해서는 주가 하락률이 다소 낮아졌지만 (A)의 경우와 (B)의 경우 모두 만족할만한 결과를 나타내지 못하였다. 다음

단계로 (C)의 경우는 (A)의 경우와 (B)의 경우를 통합하는 실험을 수행하였다.

(C)의 결과는 10k가 넘는 12개의 주식에 대해서는 19.26% 로 (A)와 (B)의 경우보다 상당히 개선되었음을 나타내고 있다. 또한, 22개의 전체 목록에 대해서도 A와 B의 경우에는 주가의 하락을 나타내었으나, (C)의 경우는 최소한 주가의 하락률이 상승률로 전환되었다.

위의 <사례연구3>을 종합하면, 주가를 예측하는데 웹 게시판의 자료를 활용하였다. 즉, 새로운 소스를 데이터마이닝³⁾에 의해 두 가지 자료를 가지고 접근하였다. 규칙기반의 학습 데이터에 기반을 두고 ‘유전자 알고리즘’ 뿐만 아니라 ‘최대 엔트로피 텍스트 분류’를 사용하였다. 각각의 예측기들을 통합하여 실험하였다. 예측 평균에 의해, 유전자 알고리즘 시도를 통해 텍스트 분석을 수행하였다.

결과적으로, <사례연구3> 또한 거래 시뮬레이션 접근법으로 판단된다. 이 연구에서 성능 평가 방법으로 개별 주식에 대한 주가의 상승률과 하락률을 사용하였다. 앞서 언급한 연구사례에 비해 가장 단순한 성능 평가 방법을 채택한 것으로 판단된다. 또한 테스트 기간이 200일에 불과하고, 22개의 한정된 주식종목에 대해 실험하였으므로, 통계적 신뢰도가 미약한 것으로 판단된다. 그러나 웹 게시판의 정보를 활용하여 주가를 예측하는 것은 유의미한 시도로 보인다. 다양한 웹 사이트 게시판의 통합적인 활용과 주가시계열의 잡음(noise)을 효과적으로 필터링 한다면 주가를 예측하는데 긍정적인 효과를 나타낼 것으로 기대된다.

위 사례연구들의 결과를 종합해보면, 대부분의 경우 거래 시뮬레이션(simulation)에 기반한 성능 평가 척도들을 채택하고 있다. 주가예측은 실제

3) 데이터 속에서 유도된 새로운 데이터 모델을 발견하여 미래에 실행 가능한 정보를 추출해내고 의사 결정에 이용하는 과정을 말한다.

벤치마크에 유용해야한다. 거래 시뮬레이션에 기반한 방식이 실제 벤치마크의 유용성을 판별하기에 적합하다. 그 이유는 거래 시뮬레이션 접근법이 현실적인 요소들을 잘 반영하고 있기 때문으로 판단된다. 또한 주식투자의 최종적인 목표는 RMSE의 최소화가 아니라 수익률(profit)의 극대화이므로 시계열 접근법보다 거래 시뮬레이션 접근법이 적합한 것으로 판단된다.

3. 기존 성능 평가 사례의 요약 및 문제점

기존의 성능 평가 방법은 시계열 접근법과 거래 시뮬레이션 접근법으로 분류되고 있다. 주가예측의 성능 평가는 시계열에 기반하여 각 척도별로 매년 도표로 나타낼 수 있다. 시계열 접근법에서 사용되는 주된 성능 평가 척도들을 표로 정리하면 다음과 같다.

시계열 접근법 (Time Series Approach)	
① 단순 예측기의 RMSE를 비교	$T_c = \frac{RMSE_y}{RMSE_{yN}}$
② 단순 수익 예측기의 RMSE를 비교	$T_r = \frac{RMSE_r}{RMSE_{rN}}$
③ ϵ -증가 예측기의 적중률을 비교	$HR_\epsilon = \frac{H_R}{H_\epsilon}$
④ 단순 수익 예측기의 적중률을 비교	$HR_N = \frac{H_R}{H_N}$
⑤ 'buy-hold'와 순이익 비교	$P_r = \frac{P_n}{P_b}$

[표 2-3 시계열 접근법의 성능 평가 척도]

시계열 접근법에서 주가예측의 방법으로 RMSE의 최소화를 채택하고 있다. 그러나 시계열 공식에 기반하여 단순히 RMSE를 최소화하는 것은 금융 시계열에서 몇 가지 예측 문제들이 있다. 그 중에서도 주가시계열의 예측에 있어서는 몇 가지 문제점을 가지고 있다. 첫째, 어떤 시간에 수행된 예측은 미래의 특정 시점에 대한 예측에 국한되어서는 안 된다. 예측 모델은 미래의 어떤 고정된 시점에 대한 예측만을 수행하는 것이 아니라 전반적인 미래 결과를 예측할 수 있어야 한다. 둘째, RMSE를 최소화한다고 해서 그것이 곧바로 수익에 직결되는 것은 아니다. 즉, 가격의 변화가 너무 작아서 비록 작은 수익을 얻더라도 거래세(transaction cost) 등의 오버헤드를 능가하지 못하는 경우, 유의미

한 예측으로 간주할 수 없다. 결국, RMSE의 최소화는 주가 예측의 확률을 높 이려는데 있지만 실질적으로 이익과 직접적인 관련성이 매우 빈약할 수도 있 다는 것이다.

주가예측에서의 성능 평가는 실제 거래에서 수익의 결과로 나타낼 수 있다. 거래 시뮬레이션 접근법이 시계열 접근법보다는 상대적으로 주가예측에서의 현실적인 요소들을 잘 반영하고 있어 실제 거래에서 수익의 결과로 나타내기 에 적합하다. 앞서 언급한 거래 시뮬레이션 접근법의 성능 평가 척도들을 표 로 나타내면 다음과 같다.

거래 시뮬레이션 접근법 (Trading Simulation)
① 전체 이익 (Total Profit)
② 일별 평균 이익 (Average Profit per Day)
③ 연간 이익 (Profit per Year)
④ 수익성 있는 거래들 (Fraction Profitable Trades)
⑤ 거래 수 (Number of Trades)

[표 2-4 거래 시뮬레이션 접근법의 성능 평가 척도]

주식 투자의 직접적인 목적은 RMSE의 최소화가 아니라 이익의 극대화이 다. 그러므로 앞서 언급한 사례연구들은 비교적 현실적인 요소를 잘 반영한 거래 시뮬레이션 접근법을 채택하고 있다. 거래 시뮬레이션 접근법은 실제 거 래상황에서 나타난 결과를 가지고 성능 평가 방법을 수행하였다. [표 2-4]에 제시된 척도들을 거래 시뮬레이션의 성능 평가 척도로 채택하고 있다.

기존의 사례연구에서는 일반적으로 연간 수익률(Profit per year) 등만을 고 려한 매우 단순한(simple) 성능 평가 척도들을 채택하고 있다. 그러나 단순한 성능 평가 척도들로는 변동성이 심한 주식 시장을 반영하는데 한계가 있을 것 으로 판단된다. 그러므로 주가 예측시스템의 성능을 개선하기 위해 추가적인

성능 평가 척도를 고려해야 할 것으로 판단된다. 다음 3장에서는 주가예측에서 고려해야할 추가적인 요소와 성능 평가 척도를 “NNShell”이라는 시뮬레이션 도구를 통해 소개하고자 한다.

III. NNShell의 성능 평가 모형

1. 평가 모형의 개요

앞서 언급한 시계열 접근법에서의 주가예측은 주식시장의 현실적인 요소들을 반영하지 못하고, 단지 예측율만을 높이려는 많은 시도들이 있었다. 시계열 접근법에서 주가 예측율을 높이려는 일반적인 방법으로는 RMSE의 최소화가 가장 대표적인 것으로 판단된다. 그러나 주식투자의 직접적인 목적은 높은 수익률이다. 그런데 RMSE의 최소화가 수익에 직결되는지에 대한 문제점이 지적되고 있다. 주가예측을 통해 직접적으로 높은 이익실현을 성취하려면 주식시장에서의 실제적인 상황을 고려해야 할 것이다. 그러므로 본 논문에서는 주식시장의 실제적인 상황과 밀접한 연관이 있는 주가예측에서 고려해야할 추가적인 요소를 반영한 “NNShell”의 성능 평가 모형을 소개하고, 아울러 NNShell에 실제 데이터에 적용한 실험결과의 분석을 통해 이 모형이 채택하고 있는 성능 평가 척도의 의미를 구체적으로 파악할 수 있는 사례를 제시하고자 한다[10].

우선 NNShell이 채택하고 있는 주가예측에서 고려해야할 추가적인 요소들을 소개한다. NNShell에서는 첫 번째로 예측 파라미터(parameter)에 해당하는 예측치들의 통계적 유의성을 보다 효과적으로 검토하기 위해, 거래 신호(signal)의 생성을 위한 다양한 예측 임계치(threshold)들을 고려하고 있다. 기존의 주가예측 연구에서는 대부분 단일 임계치에 근거하여 거래 신호를 생성하는 단순 방식을 채택하였다. 두 번째로 주식의 유동성을 고려해야할 필요가 있다. 투자자가 매수한 주식이 현재 높은 가치를 나타내고 있더라도, 보유하고

있는 것만으로는 직접적인 이익실현으로 볼 수 없을 것이다. 주식의 유동성은 '주식이 현금과 동일하게 시장에서 유통되는 상태'로 정의할 수 있다. 여기에서 현금과 완전하게 동일하다는 것은 주식을 현금으로 전환하는데 비용과 시간이 소요되지 않는 것을 의미한다. 비용발생에 따른 가치손실을 유발하지 않고 주식을 현금으로 전환할 수 있기 위해서는 무엇보다 거래가 활발하게 이루어져서 원하는 시점에서 신속하게 현금으로 전환할 수 있어야 하며, 신속한 거래를 위한 비용을 부담하지 않아야 할 것이다. 시장의 유동성을 대표하는 지표로서 거래량, 거래대금, 회전율, 거래건수 등이 있다. 유동성 지표를 아래 표에 같이 나타낼 수 있다[8].

① 거래량 및 거래대금	주식거래량은 일정기간에 매매된 개별 종목의 주식수량 거래대금이 거래량에 비해 실세를 잘 나타냄, 시장의 에너지를 알기위해 거래대금 이용
② 회전율	회전율은 일정한 기간동안(월간, 연간) 상장된 주식 중에서 얼마만큼의 주식이 거래되고 있는지를 나타냄
③ 거래건수	일정 기간동안에 매매 체결된 건수를 말한다.

[표 3-1 유동성 지표]

NNShell에서는 [표 3-1]에 제시된 거래대금과 거래건수를 고려하고 있으므로 주식의 유동성을 파악할 수 있을 것으로 판단된다. 세 번째로 NNShell에서는 매수할인율을 고려하고 있다. 주식시장은 변동성(Volatility)이 매우 심하므로 현재 시가 몇 %에 매수하느냐에 따라 수익률이 달라진다. 변동성이란 주식 수익률의 표준편차로서, 가격 또는 시장의 변동 즉, 불확실성을 나타낸다 [8]. 낮은 가격에 매수하여 높은 가격에 매도하는 것은 수익률의 기본일 것이다. 그러나 주식시장의 심한 변동성 때문에 생각처럼 쉬운 문제는 아니다. 계

속적으로 상승세가 기대되는 주식은 현재 시가보다 높은 가격에 매수하여도 수익을 올릴 수 있다. 그러나 주가가 상승할 것을 기대하고 현재시가보다 높게 매수한 종목이 하락하면 손실을 기록하게 된다. 매수할인율의 의미를 파악하기 위해 아래 도표를 이용하여 구체적인 실례를 제시한다[9].



[그림 3-1 주가 차트]

투자자는 A시점에 20%의 기대 수익률을 가지고 520원에 개별 종목을 매수하였다. 만약 B시점 580에 매도하면 약 12%의 수익률을 올려 기대수익에 못 미치는 시점에서 매매가 이루어진다. 그러나 C시점에 620원이 넘는 가격으로 매도하면 목표 기대 수익률인 20% 이상의 수익을 성취할 수 있다. 마찬가지로 B시점에 매수하여 C시점에 매도하여도 기대 수익률에 도달하지 못한다. 또한 C시점에서 계속적으로 주가가 상승할 것 기대하여 현재의 종가보다 높은 가격에 매수하였으나, 주가가 하락세로 전환되어 D시점에서 10%가 넘는 손실을 기록하며 매도하였다. 이와 같이 어느 시점에 매수하여 매도하느냐에 따라 수익률을 달라진다. 네 번째로 NNShell에서는 이익실현율(profit ratio)을

고려하고 있다. 실제 주식시장에서 일반적인 투자방식은 가격 상승이 기대되는 개별 종목을 매수하고, 기대치만큼 상승하면 매도하여 이익을 실현시킨다. NNShell에서는 투자자의 기대수익의 수치를 나타내는 이익실현율을 고려함으로써 효율적인 투자전략 수립이 가능할 것으로 보인다. 다섯 번째로 NNShell에서는 손절율(loss ratio)을 고려하고 있다. 투자자들은 가격 상승이 기대되는 주식을 매수하지만 투자자들의 예측과 달리 주가가 하락하는 경우 손실이 발생하기 마련이다. 이런 경우 위험부담을 최소화하기 위해 손절율을 고려하는 것이다. NNShell에서는 기존 사례연구에서 제시하지 못했던 주가예측에서 고려해야 할 추가적인 요소들을 앞서 소개한 바와 같이 반영하고 있다.

앞서 언급한 사례연구들은 대부분 거래 시뮬레이션 기반으로 시계열 접근법 보다는 주식시장의 현실적인 요소를 잘 반영하고 있으나, 연간 수익률만을 고려한 매우 단순한 성능 평가 방법을 채택하고 있다. NNShell은 기존 사례연구에 비해 다양한 성능 평가 척도들을 채택하고 있는 것으로 보인다. NNShell이 채택하고 있는 성능 평가 척도들은 다음과 같다.

첫 번째로 NNShell은 성능 평가 척도로 적중률을 채택하고 있다. 주가 예측 시스템에서 주가가 상승할 것으로 판단되는 개별 종목을 추천하면, 실제거래에서 어느 정도의 성공률 나타내는가의 판단 기준이 된다. 적중률은 성공적인 거래횟수/추천수로 나타낼 수 있다. 적중률은 주가 예측시스템의 정확도를 직관적으로 나타낼 수 있는 중요한 성능 평가 척도이다. 두 번째로 거래당 수익률을 성능 평가 척도로 채택하고 있다. 거래당 수익률은 전체이익/거래횟수로 표현할 수 있다. 거래당 수익률은 성능 평가의 효율성을 판단할 수 있는 지표가 될 수 있다. 즉, 같은 100만원의 수익을 올렸더라도 몇 번의 거래횟수냐에 따라 거래당 수익률에서 차이를 나타낸다. 예를 들면, 50번의 거래로 100만원 수익을 올린 경우의 거래당 수익률은 2만원이고, 100번의 거래로 100만원 수

익을 올린경우의 거래당 수익률은 1만원이다. 주식 거래에 대한 거래세까지 고려한다면 거래당 수익률은 반드시 필요한 성능 평가 척도이다. 거래당 수익률에서 한 가지 요소를 추가적으로 고려한다면 주식 보유일수를 들 수 있다. 거래당 수익률을 거래당 수익률/보유일수로 나타내면 하루의 평균수익을 산출할 수 있다. 주식 보유일수를 고려함으로써 하루평균 수익에 따른 성능 평가를 할 수 있다. 마지막으로 NNShell에서는 성능 평가 척도로 수익금을 채택하고 있다. 실제 주식투자의 최종적인 목표는 높은 수익률 성취에 있으므로, 가장 직관적인 성능 평가 척도로 판단된다. 앞서 소개한 NNShell의 성능 평가 척도들은 주가 예측시스템의 성능을 개선시킬 것으로 추측된다.

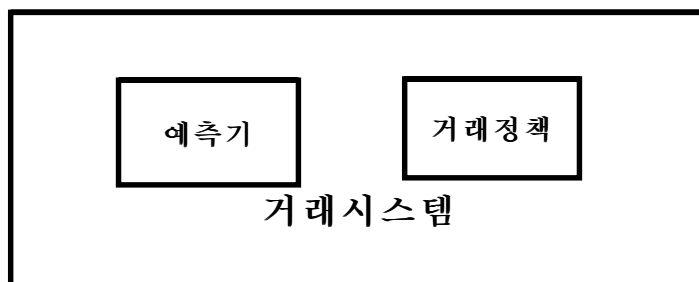
앞서 소개한 NNShell이 채택하고 있는 요소들 중에는 거래정책 해당하는 인자들이 있다. 주가예측과 거래정책을 통합하면 효율적인 투자전략 수립이 가능할 것으로 판단된다. 그러므로 다음 2절에서는 주가예측과 거래정책의 통합 시뮬레이션의 개념을 소개한다.

2. “주가예측 - 거래정책” 통합시물레이션



[그림 3-2 지능형 거래시스템의 일반적인 구조]

기존의 지능형 거래시스템의 일반적인 구조는 [그림 3-2]와 같이 예측기가 수익성이 기대되는 주식의 개별 종목을 추천만 하였을 뿐 거래정책은 독립적으로 이루어졌다. 주가예측에서는 각 개별 종목에 해당하는 예측치가 있다. 각각의 예측치에 대해 임계치를 고려할 수 있다. 그러나 기존의 예측기는 예측 파라미터에 해당하는 단일 임계치에 근거하여 주가 예측을 수행한 것으로 판단된다. 주가 예측시스템에서 예측율은 주식거래에 대한 예상수익률의 수치를 나타낸다. 투자자들이 효율적인 투자전략을 수립하고 예측율과 실제수익의 상관성을 파악하기 위해 주가 예측시스템은 주가예측과 거래정책이 통합적으로 구성되어야 할 것으로 판단된다. 일반적으로 거래시스템의 통합구조는 [그림 3-3]과 같이 나타낼 수 있다.



[그림 3-3 주가예측과 거래정책의 통합 구조]

주가예측-거래정책의 통합구조는 다양한 예측 임계치에 근거하여 주가예측을 수행하며 거래정책을 거래시스템에 포함하였다. 거래정책 인자는 앞서 소개한 매수할인율, 이익실현율, 손절율 등이 해당된다. 주가예측과 거래정책이 통합된 사례로 연구[11]에 의하면 ‘지역적 정책’이라고 불리는 ‘LP(local policy)’를 사용하는 시뮬레이션 기법을 채택한 사례가 있다. LP는 다음과 같은 기준(metrics)으로 거래정책을 통합하였다. 공간 $\Omega = (B_THRES, A_THRES, H_DURATION)$ 이 주어졌을 때 LP는 각 예측기의 최적화된 파라미터들의 벡터이다. 즉, 지역정책 LP는 LP(predictor) = (b_thres, a_thres, h_duration)⁴⁾으로 나타낼 수 있다. 여기서 중요한 사실은, 두 임계치 (b_thres, a_thres)는 예측 파라미터에 해당하는 수치이고, 보유기간은 정책 파라미터인데, 이들 파라미터들이 통합 시뮬레이션에 의해 미리 결정된 값들이라는 것이다. LP의 수행과정은 우선 b_thres보다 예측치가 큰 주식들이 검색되어 매수된다. 이 주식들은 예측치가 a_thres보다 작아지거나 보유일이 만기될 때 매도된다. 각 예측기의 LP를 최적화하기 위해 거래당 수익률과 정확도를 측정기준으로 사용하였다. 위 연구의 ‘지역적 정책 LP’는 주가예측과 거래정책을 통합한 사례이기는 하나, 단순히 매수 임계치와 매도 임계치만을 고려하였을 뿐이다. LP는 단순한 형태의 통합정책⁵⁾을 채택하고 있다. 단순한 형태의 통합정책은 복잡한 주식시장에서 실제 거래상황을 반영하기 어려울 것으로 판단된다. 그러므로 앞서 소개한 거래정책 인자들을 거래시스템에 포함하면 거래시스템의 성능이 개선될 것으로 추측된다. “주가예측-거래정책”의 통합기법을 NNShell은 채택하고 있으며, 또한 “지역 정책 LP”의 거래정책 요소 이외에도 매수할인율, 이익실현율, 손절율, 보유기간 등의 다양한 거래정책 요소를 반영하고 있다.

4) b_thres : 매수 신호의 임계치, a_thres: 매도 신호의 임계치, h_duration: 최대 주식 보유기간

5) 주가예측과 거래정책의 통합을 통합정책이라 한다.

3. NNShell을 이용한 성능 평가 분석 사례

앞서 나열한 NNShell의 평가 요소6)와 성능 평가 척도7)의 최적화 의미를 파악해보기 위해 실제 데이터를 적용한 시뮬레이션 수행결과를 예시하기로 한다. 첫 번째는 NNShell에서는 예측치 베이스로 임계치를 고려하여 주가 예측을 수행하였다. 예측치 베이스가 0.4와 0.5인 경우 그리고 예측치 베이스가 0.6인 경우 세 가지의 경우를 NNShell 펀드 시뮬레이션을 수행하였다.

1) 예측치 베이스가 0.4 인 경우



[그림 3-4 NNShell의 시뮬레이션 수행예제1]

예측치 베이스가 0.4인 경우, 결과는 959번의 매매로 수익금이 636,893,387

- 6) 평가 요소란 임계치와 매수할인율, 이익실현율, 손절율 등의 거래정책 인자로서 성능 평가 척도와 구분된다.
- 7) 성능 평가 척도는 적중률, 거래당 수익률, 수익금 등으로 주가 예측시스템의 성능을 측정할 수 있는 것을 말한다.

원, 적중률은 50.3%, 거래당 수익률은 10.5%, 보유일수 7.03일, 손절매는 128번으로 나타났다. 예측치 베이스가 0.4인 경우는 매매횟수가 959번으로 상당히 많은 거래가 이루어지고 있고, 이로 인해 높은 수익금을 나타내고 있다. 그러나 적중률은 50.3%로 비교적 낮은 결과를 보이고 있다. 예측치 베이스가 0.4인 경우와 비교 분석하기 위해 예측치 베이스 수치를 높여 NNShell 펀드 시뮬레이션을 수행하였다.

2) 예측치 베이스가 0.5인 경우

예측치 베이스 수치를 0.4에서 0.5로 높여 NNShell 펀드 시뮬레이션을 수행하였다. 수행결과는 아래 [그림 3-5]와 같다.



[그림 3-5 NNShell의 시뮬레이션 수행예제2]

예측치 베이스가 0.5인 경우의 결과는 451번의 매매로 수익금이 364,365,665 원, 적중률은 59.2%, 거래당 수익률은 12.74%, 보유일수는 6.56일, 손절매는 45 번으로 나타났다. 예측치 베이스가 0.4인 경우와 0.5인 경우를 비교하면, 수익 금은 636,893,387원에서 364,365,665원으로 상당히 감소하였다. 이 결과는 예측 지 베이스를 높임으로써 주가 상승이 예상되는 확실한 종목에만 투자하게 되 므로 매매횟수의 감소로 수익금이 감소하였다. 그러나 적중률이 50.3%에서 59.2%로 높아졌고, 거래당 수익률 또한 10.5%에서 12.74%로 높아졌다. 예측치 베이스가 높을수록 적중률과 거래당 수익률이 높아짐을 알 수 있다. 예측치 베이스가 0.5보다 큰 경우인 0.6으로 하여 NNShell 펀드 시뮬레이션을 수행하 여 세 가지의 경우를 비교 분석하였다.

3) 예측치 베이스가 0.6인 경우



[그림 3-6 NNShell의 시뮬레이션 수행예제3]

예측치 베이스가 0.6인 경우, 결과는 175번의 매매로 수익금이 169,694,910 원, 적중률은 69.1% 거래당 수익률은 15.74%, 보유일수 5.97일, 손절매는 14번으로 나타났다. 예측치 베이스가 0.6인 경우는 예측치 베이스가 0.4와 0.5인 경우에 비해 거래횟수가 175번으로 매우 감소한 결과를 보이고 있다. 수익금 또한 0.4인 경우와 비교하면 1/6 가량 감소한 결과를 나타내고 있다. 그러나 예측치 베이스가 0.6인 경우는 69.1%의 높은 적중률이 나타나고 있다. 또한 예측치 베이스를 높여감에 따라 손절매 횟수가 감소하는 결과를 나타내고 있다. 즉, 예측치 베이스를 다르게 설정하여 실험한 결과를 종합해보면, 예측치 베이스의 수치가 높은 경우, 확실히 수익이 기대되는 개별 종목에만 투자하므로 매매횟수를 감소시켰다. 예측치 베이스가 높아짐에 따라 거래횟수가 줄어들고 그 결과로 수익금이 감소하였다. 반면에 예측치 베이스가 높아짐에 따라 적중률과 거래당 수익률은 최적화된 결과를 나타낸다. 예측치 베이스가 높아짐에 따라 매매횟수의 감소로 수익금은 감소하였지만 확실히 수익이 기대되는 개별 종목에만 투자하기 때문에 손절매 횟수가 감소한다. 그러므로 안정적인 투자가 가능하다.

두 번째로 NNShell은 주가예측과 거래정책이 통합된 시뮬레이션 도구로써 거래정책 요소인 매수할인율을 고려하였다. 앞서 언급한 것처럼 주식시장은 변동성이 심하므로 매수할인율에 따라 수익률이 달라질 수 있다. 모든 조건이 동일한 경우 매수할인율이 다른 경우의 성능 평가 척도인 수익금을 비교하였다.

조 합	매수 할인율	이익 실현율	손절율	매매횟수	보유일수	수익금
(0,8,0) × (0,4,9)	1.06%	1.24%	0.82%	959	9	636,893,387원
(0,2,0) × (0,4,9)	0.98%	1.24%	0.82%	578	9	376,340,404원

[표 3-2 매수할인율에 따른 수익금 변화]

매수할인율이 1.06%인 경우의 수익금은 636,893,387원으로 나타났고, 매수할인율이 0.98%인 경우의 수익금은 376,340,404원으로 나타났다. 매수할인율이 높은 지점보다 낮은 지점에서 매매횟수의 감소로 수익금이 감소하였다. 이 결과를 분석해보면 예측력이 좋은 엔진의 경우는 매수할인율이 높은 지점에서 매매가 이루어지는 것도 무방할 것으로 판단된다. [표 3-2]에서 매수할인율이 높은 지점에서 매매횟수가 959번이었지만 매수할인율이 낮은 지점에서는 578번으로 매매횟수가 감소하였고 이로 인해 수익금이 감소하였다. 예측력이 낮은 엔진의 경우는 매수할인율이 낮은 지점에서 매매가 이루어지는 것이 바람직할 것으로 보이고, 이 경우 거래횟수 또한 감소한다.

세 번째로 NNShell에서 이익실현율 구간을 [24, 22, 20, 18, 16, 14, 12, 10]으로 선택하여 NNShell 펀드 시뮬레이션을 수행하였다. 거래정책 요소인 이익실현율에 따른 적중률과 수익금을 비교 분석하였다.



[그림 3-8 NNShell의 시뮬레이션 수행예제5]

NNShell 펀드 시뮬레이션 수행결과는 [표 3-3]과 같다. 이익실현율이 낮은 지점에서 최고의 적중률 79.6%로 나타났다. 이익실현율이 높은 지점에서 적중률은 56.6%로 이익실현율이 낮은 지점보다 상대적으로 낮게 나타나고 있다.

조합	매수 할인율	이익 실현율	손절율	매매 횟수	보유 일수	적중률	수익금	거래당 수익률
(0,1,0) × (7,4,9)	0.96%	1.10%	0.82%	426	9	79.6%	121,516,924원	4.33%,
(0,1,0) × (1,4,9)	0.96%	1.22%	0.82%	426	9	56.6%	278,457,896원	9.92%

[표 3-3 이익실현율에 따른 수익금의 변화]

모든 조건이 동일하고 거래정책 요소인 이익실현율에 다르게 설정하였을 경우의 수행결과는, 이익실현율이 낮은 지점에서는 높은 적중률의 결과를 나타내고 있다. 이익실현율이 10%까지 도달하면 거래가 이루어지므로 높은 적중률을 보이고 있다. 그러나 이익실현율이 22%인 경우는 보유기간 동안에 이익실현율에 도달하지 못하면 거래가 성립되지 않으므로 낮은 적중률을 나타내고 있다. (0,1,0)×(1,4,9)의 조합은 낮은 적중률을 나타내고 있지만 426번의 같은 매매횟수로 (0,1,0)×(7,4,9)의 조합보다 높은 수익금을 나타내고 있다. 이 결과는 이익실현율이 높은 지점에서 거래가 이루어지기 때문에 거래당 수익률이 높아지기 때문으로 판단된다. 투자자가 매수한 주식에 대해 22%의 이익실현율을 목표로 설정하였지만 주식보유 만기일 되면 기대수익에 못 미쳐도 매도하는 것을 원칙으로 한다. NNShell에서는 이익실현율을 22%로 설정하면 22% 이상 수익을 기록한 것을 이익실현율이라고 하고, 기대수익에 미치지 못하였지만 보유만기일이 되어서 매매된 거래를 '이익'으로 구분하였다.

네 번째로 NNShell에서는 거래정책 인자에 해당하는 손절을 구간을 설정할 수 있다. 손절을 구간을 [-10, -12, -14 -16, -18]로 설정하고 모든 조건이 동일하고 손절율만 다른 조합을 가질 경우의 수익금을 비교하였다.

같은 매매횟수로 손절율이 높은 경우(0.82%)의 수익금은 636,893,387원으로 나타났고, 손절율이 낮은 경우(0.90%)의 수익금은 568,126,202원으로 약간 감소하였다. 이 결과를 분석하자면, 손절율이 높은 경우는 보유기간 동안 18% 이상 하락한 경우에만 손절매를 실시하므로 손절매의 횟수가 손절율이 낮은 경우보다 적게 나타났다. [표 3-4]의 결과에서도 손절율이 높은 경우, 손절매의 횟수가 128번인데 비해 손절율이 낮은 경우, 289번으로 손절매 횟수가 증가하였다. 여기서 수익금의 차이는 손절매 횟수 때문인 것으로 판단된다. NNShell에서 손절율을 18%로 설정하면 18% 이상 손실을 기록한 개별 종목에 대해서만 손절매를 한다. 투자자가 보유한 주식의 가격이 아직 3% 밖에 하락하지 않았지만 손절매를 한다. 투자자가 보유한 주식이 아직 3% 밖에 하락하지 않았지만 손절매를 한다. 투자자가 보유한 주식이 아직 3% 밖에 하락하지 않았지만 손절매를 한다.

마지막으로 거래대금의 상한값과 하한값을 다르게 설정하여 수익금의 변화 여부를 NNShell 펀드 시뮬레이션을 수행하여 비교 분석하였다.

1) 거래대금 상하한 5억~20억인 경우



[그림 3-10 NNShell의 시뮬레이션 수행예제 7]

거래대금⁸⁾의 상한값을 20억으로 제한하여 NNShell의 펀드 시뮬레이션을 수행하였다. 그 결과는, 655번의 매매로 수익금은 335,942,728원을 나타냈다. 거래대금의 상한값이 수익금의 변화에 어떤 영향을 미치는지 알아보기 위해 거래대금을 상향 조정하여 NNShell 펀드 시뮬레이션을 수행하여 비교 분석하였다.

2) 거래대금 상하한 5억~50억인 경우

모든 조건이 위와 동일하고 거래대금의 상한값을 20억에서 50억으로 상향 조정하여 NNShell 펀드 시뮬레이션을 수행하였다.



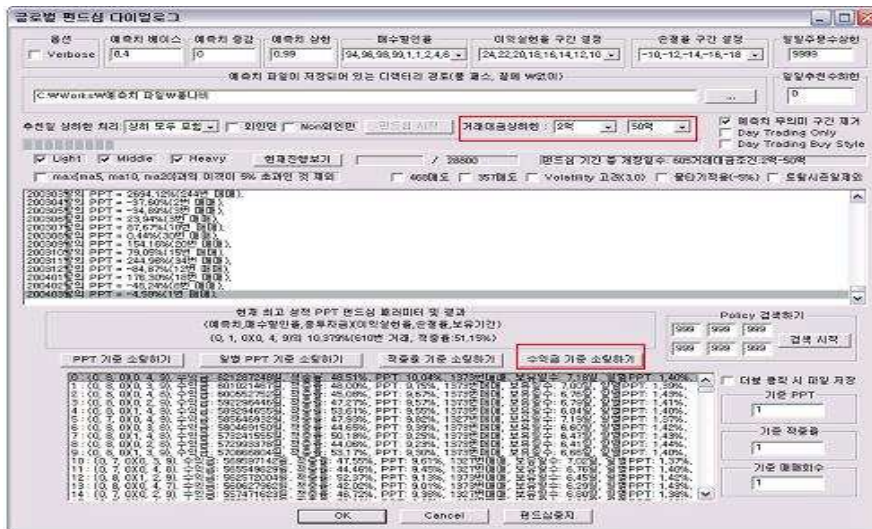
[그림 3-11 NNShell의 시뮬레이션 수행예제8]

8) 거래대금이란 하루 동안에 해당주식이 거래된 금액을 말한다. 최근 20거래일 동안의 거래대금의 평균을 일일 거래대금으로 간주하였다.

그 결과는 850번의 매매로 수익금은 542,069,794원을 나타냈다. 거래대금을 20억에서 50억으로 상향 조정한 결과, 거래횟수가 655번에서 850번으로 증가하였고 이로 인해 수익금이 증가하였다. 거래대금이 20억인 경우는 자금의 한계로 인해 투자에 제한이 있었기 때문에 비교적 적은 거래횟수를 나타내고 있다. 그러나 거래대금이 50억인 경우는 보다 많은 개별종목에 투자할 수 있으므로 거래횟수가 증가하였고 이로 인해 수익금이 증가한 결과를 나타내고 있다. 거래대금의 상한값이 큰 경우와 작은 경우, 수익금의 변화를 비교 분석하였다. 거래대금의 하한값도 수익금의 변화에 영향을 주는지 알아보기 위해 거래대금의 하한값을 하향 조정하여 NNShell 펀드 시뮬레이션을 수행하였다.

3) 거래대금 상하한 2억-50억인 경우

거래대금의 하한값을 5억에서 2억으로 하향 조정하여 NNShell 펀드 시뮬레이션을 수행하였다.



[그림 3-12 NNShell의 시뮬레이션 수행예제9]

수행결과는 1373번의 매매로 수익금은 621,287,248원으로 나타났다. 거래대금의 하한을 5억에서 2억으로 하향 조정한 결과는 매매횟수가 상당히 증가하였고 이로 인해 수익금이 증가하였다. 이는 거래대금의 하한값을 낮춤으로써 작은 거래들까지 포함하므로 매매횟수가 증가하였고, 이로 인해 수익금이 증가하였다. 거래대금의 상하한에 따른 실험 결과를 종합해보면 거래대금의 상한값이 작은 경우에는 투자에 제한이 따른다. 즉, 투자자들은 자금의 부족으로 인해 추천이 난 모든 종목에 투자할 수 없을 것으로 보인다. 그리고 거래대금의 하한값이 작은 경우는 수익이 적은 거래를 모두 포함하게 되므로 거래횟수가 상당히 증가하였다.

IV. 추가적인 성능 평가 척도의 제안

NNShell은 기존 추가 예측시스템에 비해 다양한 성능 평가 척도를 채택하고 있으며, 추가예측에 거래정책을 포함한 통합 시뮬레이션 도구이다. 3장에서 성능 평가 척도들과 거래정책 요소들의 최적화 의미를 파악하기 위해 NNShell 펀드 시뮬레이션을 수행하였다. 앞서 소개한 추가예측에서 고려해야 요소들은 거래정책 인자와 성능 평가 척도로 아래 표와 같이 구분할 수 있다.

성능 평가 척도	거래 정책 인자
$\text{적중률} = \frac{\text{성공적인거래횟수}}{\text{추천수}}$	매수할인율
$\text{거래당수익률} = \frac{\text{전체이익}}{\text{거래횟수}}$	이익실현율
$\text{일별PPT} = \frac{\text{거래당수익률}}{\text{보유일수}}$	손절율
수익금	보유기간

[표 4-1 NNShell의 성능 평가 척도 및 거래정책 요소]

제 3장에서는 성능 평가 척도들의 최적화 의미를 파악하기 위해 NNShell 펀드 시뮬레이션을 수행하였다. 본 4장에서는 NNShell 시뮬레이션 도구의 성능 평가 척도들을 분석하기 위해 시뮬레이션의 수행결과를 비교 분석하였다.

<거래예제1>

080049-0.220-0.000-0.990-예측치0.220이상 5억-1조 1.06매수 1.24이익실현 0.82손절 9일보유.txt
 수익금: 571330121원. 보유일수: 6.80일, 일별PPT: 1.46%, 매매회수:862, PPT:9.91%,이익실현:456(52.9%),
 이익:158(18.3%), 손절:142(16.5%), 손실:106(12.3%), 적중률:52.9%

038120 프로소닉# *20020123일 0.23649 20020124일 2810원 20020130일 2080원 -26.779 8756407
 038120 프로소닉# *20020201일 0.22875 20020204일 1750원 20020220일 1480원 -16.229 17377053
 038120 프로소닉# *20020204일 0.43571 20020205일 1650원 20020221일 1480원 -11.103 17758253
 038120 프로소닉# *20020205일 0.48429 20020206일 1600원 20020222일 1430원 -11.425 17791080
 038120 프로소닉# *20020206일 0.30087 20020207일 1770원 20020219일 1451원 -18.800 20421734
 031310 아이즈비전# *20020208일 0.34294 20020214일 1210원 20020218일 1500원 23.200 15108349
 038120 프로소닉# *20020208일 0.25696 20020214일 1560원 20020227일 1420원 -9.774 15610445
 002780 진흥기업 *20020208일 0.24145 20020214일 1570원 20020225일 1287원 -18.800 17453672
 002780 진흥기업 *20020214일 0.37033 20020215일 1505원 20020226일 1234원 -18.800 16628662
 002780 진흥기업 *20020215일 0.42994 20020218일 1400원 20020304일 1185원 -16.157 16736780
 037700 인디시스템# *20020215일 0.24153 20020218일 5340원 20020304일 6570원 22.234 19608983
 002780 진흥기업 *20020218일 0.48500 20020219일 1390원 20020305일 1205원 -14.109 17069750
 037700 인디시스템# *20020218일 0.24602 20020219일 5600원 20020305일 6944원 23.200 19682081
 000660 하이닉스 *20020221일 0.32019 20020222일 1505원 20020305일 1866원 23.200 2439566783
 017050 스타맥스# *20020221일 0.22840 20020222일 1600원 20020307일 1984원 23.200 5684405
 000660 하이닉스 *20020222일 0.27966 20020225일 1550원 20020305일 1922원 23.200 2384888839

.....

200202	-26.78	1번	매매
200202	58.24%	21번	매매
200203	-56.33%	6번	매매
200204	174.14%	51번	매매
200205	320.25%	36번	매매
200206	2915.21%	172번	매매
200207	761.24%	54번	매매
200208	-60.57%	13번	매매
200209	104.88%	62번	매매
200210	622.88%	42번	매매
200211	164.08%	11번	매매
200212	159.69%	10번	매매
200301	33.69%	11번	매매
200302	4.40%	2번	매매
200303	2874.40%	194번	매매
200305	-1.45%	1번	매매
200307	-32.21%	2번	매매
200308	60.59%	15번	매매
200309	69.74%	23번	매매
200310	-149.83%	20번	매매
200311	200.99%	27번	매매
200312	-21.62%	9번	매매
200401	106.06%	18번	매매
200402	12.97%	3번	매매
200403	0.74%	8번	매매
200405	83.53%	23번	매매
200406	159.93%	27번	매매

[표 4-2 NNShell의 시뮬레이션 결과파일 1]

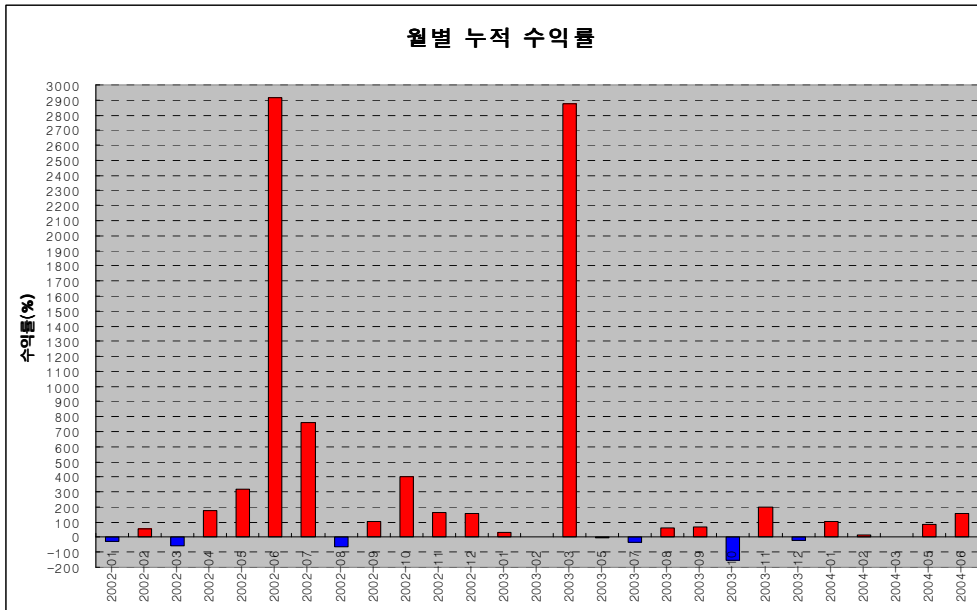
1) <거래 예제1>의 시물레이션 결과

시물레이션 수행조건	시물레이션 수행결과
예측치: 0.22이상	수익금: 571,330,121원 매매횟수: 862번
거래대금 상하한: 5억 -1조	평균보유일수 : 6.80일
매수할인율 :1.06	거래당 수익률 : 9.91 일별PPT:1.46%
이익실현율 구간:1.24	적중률: 52.9% 이익실현율 52.9%
손절율 구간: 0.82	손절율: 16.5%
최대 보유일: 9일	이익: 18.3% 손실: 12.3%

[표 4-3 NNShell의 시물레이션 수행 결과 1]

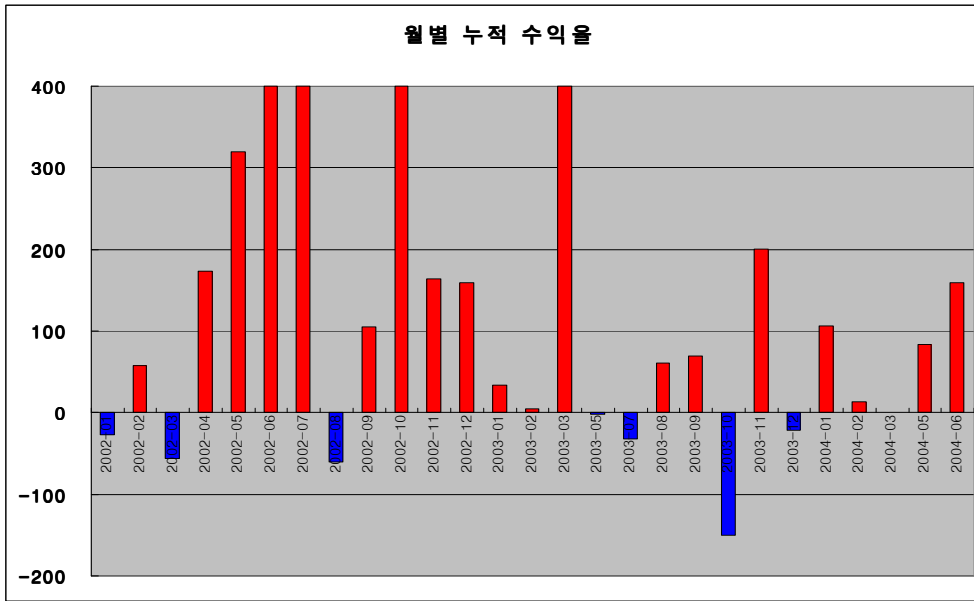
<거래예제1>의 시물레이션 결과를 분석해보면, 거래당 수익률과 수익금이 비교적 높게 나타났지만 월별 수익률⁹⁾이 고르게 분포하지 않은 것을 알 수 있다. 월별 수익률의 분포를 살펴보면 2002년 6월과 2003년 3월에 특정한 기간에 20배가 넘는 수익률을 나타내고 있다. 앞서 3장에서 거래대금의 제한으로 투자에 제한이 있을 수 있음을 NNShell 펀드 시물레이션 결과를 통해 알 수 있었다. <거래예제1>의 월별 수익률을 그래프를 통해 나타내고 분석하였다. [그림 4-1]을 분석하면 특정기간에 집중적인 월별 수익률을 보이고 있으며, 전반적으로 월별 수익률이 고르게 분포하고 있지 않음을 나타낸다.

9) 월별 수익률이란 한 달 동안의 거래를 통해 누적된 수익률을 말한다.



[그림 4-1 <거래예제1>의 월별 수익률]

실제 주식시장에서 투자자들은 자금의 한계로 투자에 제한이 따를 것으로 판단된다. 가장 큰 월별 수익률을 나타내고 있는 2002년 6월과 2003년 3월의 거래횟수는 172번과 194번으로 다른 기간에 비해 상대적으로 많다. 투자자금의 한계로 현실적으로 많은 거래가 다 이루어지지 못할 것으로 예상된다. 이에 본 논문에서는 한 가지 가설을 설정한다. 최대 월별 수익률을 400%로 제한한다. 최대 월별 수익률을 400%로 제한하여 그래프로 나타내면 [그림 4-2]와 같다.



[그림 4-2 <거래예제1>의 가설 그래프]

[그림 4-2]를 분석하자면, 일정한 기간의 비슷한 패턴을 보이고 있으며, 특정한 기간에 집중적인 예측력을 보이고 있다. 월별 수익률의 최대 한계점까지 도달한 경우가 여러 번 나타나고 있다. <거래예제1>의 높은 이익실현율은 전반적으로 고른 거래 때문이 아니라 특정한 기간의 집중적인 수익률 때문인 것으로 그래프를 통해 파악할 수 있다. 이는 주가 예측시스템이 특정한 기간에 집중적인 예측력을 보인 것으로 판단된다. 그러나 여기서 우리는 한 가지의 구심을 가질 수 있다. 그 당시의 경제상황과 관계없이 단지 주가 예측시스템의 순수한 예측력만으로 특정한 기간에 집중적인 예측력을 보였다 할 수 있을 것인가? 즉, 주가 예측시스템의 순수한 예측력만으로 특정한 기간에 집중적인 예측력인 보인 것인지 아니면 그 당시의 일시적인 주식시장의 호황 때문인지를 판단할 수 있어야 한다. 특정한 기간에만 집중적인 예측력을 보이는 주가 예측시스템은 투자자들에게 신뢰감을 주기 어려울 것으로 보인다. 이에 투

투자자들의 투자 심리를 고려한 거래시스템의 구성이 요구된다. 이에 본 논문에서는 투자자의 투자 심리를 고려한 월별 표준편차와 월별 적중률을 거래시스템에 제안한다. 실제 주식투자에서 거래의 안정성을 파악하기 위한 판단지표로 월별 표준편차를 제안한다. [그림 4-1]의 월별 수익률을 나타낸 도표를 분석하면 어느 특정한 달에 20배가 넘는 월별 수익률의 결과를 나타내고 있지만 실제 주식시장에서는 자금의 한계로 투자의 제한 있을 것으로 보인다. 월별 표준편차를 거래시스템에 포함하여 성능 평가 척도로 채택하면 보다 효율적인 방법으로 월별 수익률의 변동을 파악할 수 있다. 월별 표준편차를 공식으로 나타내면 다음 <정의1>과 같다.

<p><정의1> 월별 표준편차</p> $\frac{\sqrt{\sum_{i=1}^n (r_i - \bar{r}_i)^2}}{N}$ <p>r_i: 월별 수익률, \bar{r}_i: 평균 월별수익률, N: 총 개월수</p>

위 공식에 적용하여 <거래예제1>의 월별 표준편차를 산출하면 는 367이다. 월별 표준 편차가 매우 높게 나타났으므로 객관적인 지표로서 월별 수익률의 변동이 매우 심함을 나타낸다. 월별 표준편차가 크면 클수록 투자자의 심리에 불안요소로 작용할 것으로 판단된다. 또한 월별 수익률의 손실을 기록한 달 또한 여러 번 나타나고 있다. 이는 투자자의 심리에 영향을 미칠 것으로 예상되므로 전체 거래에 대한 객관적인 지표로 월별 적중률을 제안한다. 월별 적중률을 공식으로 나타내면 <정의2>와 같다.

<p><정의2> 월별 적중률</p> $\left \frac{r_i > 0}{N} \right $ <p>r_i: 월별 누적수익률, N: 총 개월 수</p>
--

<거래예제1>의 월별 적중률은 74%이다. 월별 적중률은 전체적인 거래기간 동안의 판단 지표로 투자자들에게 신뢰감을 줄 수 있는 성능 평가 척도로 사용될 수 있을 것으로 판단된다. 월별 적중률은 74%로 주가 예측시스템의 예측력은 비교적 높다고 볼 수 있으나, 월별 표준편차가 상대적으로 크기 때문에 월별 수익률의 변동이 매우 심하다. 이는 투자자의 심리에 불안요소로 작용할 것으로 보인다.

<거래예제2>

080545-0.290-0.000-0.990-예측치0.290이상 5억-1조 1.06매수 1.14이익실현 0.82손절 5일보유.txt:
 수익금: 105395092원.보유일수: 3.90일, 일별PPT: 2.08%, 매매회수:172, PPT:8.11%, 이익실현:123(71.5%),
 이익:18(10.5%), 손절:9(5.2%), 손실:22(12.8%), 적중률:71.5%

038120	프로소닉#	*20020204일	0.43571	20020205일	1650원	20020207일	1880원	13.200	17758253
038120	프로소닉#	*20020205일	0.48429	20020206일	1600원	20020207일	1823원	13.200	17791080
038120	프로소닉#	*20020206일	0.30087	20020207일	1612원	20020219일	1600원	-1.544	20421734
031310	아이즈비전#	*20020208일	0.34294	20020214일	1210원	20020215일	1379원	13.200	15108349
002780	진흥기업	*20020214일	0.37033	20020215일	1447원	20020220일	1649원	13.200	16628662
002780	진흥기업	*20020215일	0.42994	20020218일	1400원	20020219일	1595원	13.200	16736780
002780	진흥기업	*20020218일	0.48500	20020219일	1390원	20020219일	1584원	13.200	17069750
000660	하이닉스	*20020221일	0.32019	20020222일	1505원	20020228일	1715원	13.200	2439566783
000660	하이닉스	*20020225일	0.36567	20020226일	1465원	20020228일	1670원	13.200	2065821606
035450	세넥스테크#	*20020228일	0.29934	20020304일	2300원	20020304일	2622원	13.200	12233680
035450	세넥스테크#	*20020306일	0.38912	20020307일	2020원	20020311일	2302원	13.200	23137981
037760	삼원정밀금속#	*20020311일	0.34444	20020312일	1290원	20020319일	1270원	-2.350	25920923
044370	로토토#	*20020411일	0.33691	20020412일	4500원	20020417일	5130원	13.200	18684595
044370	로토토#	*20020412일	0.40788	20020415일	4520원	20020417일	5152원	13.200	17684455
044370	로토토#	*20020415일	0.36333	20020416일	4300원	20020417일	4902원	13.200	19975391
009730	동신에스엔터#	*20020424일	0.33483	20020425일	1385원	20020430일	1135원	-18.800	11689010
042340	이네트#	*20020424일	0.30844	20020425일	6387원	20020503일	5680원	-11.869	27884246
039230	파이컴#	*20020424일	0.30185	20020425일	5090원	20020430일	4173원	-18.800	44498003
042340	이네트#	*20020425일	0.44585	20020426일	5640원	20020426일	6429원	13.200	27892230
009730	동신에스엔터#	*20020425일	0.44297	20020426일	1235원	20020506일	1407원	13.200	11883227
039230	파이컴#	*20020425일	0.30962	20020426일	4550원	20020503일	5187원	13.200	39825120
035270	청람디지탈#	*20020425일	0.30598	20020426일	3560원	20020502일	4058원	13.200	61265
.....									
.....									
200202			117.26%	10번	매매				
200203			10.85%	2번	매매				
200204			228.19%	26번	매매				
200205			155.36%	16번	매매				
200206			50.61%	9번	매매				
200207			13.20%	1번	매매				
200208			6.04%	3번	매매				
200209			100.37%	23번	매매				
200210			13.20%	1번	매매				
200211			39.60%	3번	매매				
200301			39.60%	3번	매매				
200302			13.20%	1번	매매				
200303			8.00%	4번	매매				
200308			68.73%	8번	매매				
200309			81.29%	13번	매매				
200310			58.23%	9번	매매				
200311			113.43%	12번	매매				
200312			-5.60%	2번	매매				
200401			113.20%	11번	매매				
200403			39.60%	3번	매매				
200405			32.67%	4번	매매				
200406			97.71%	8번	매매				

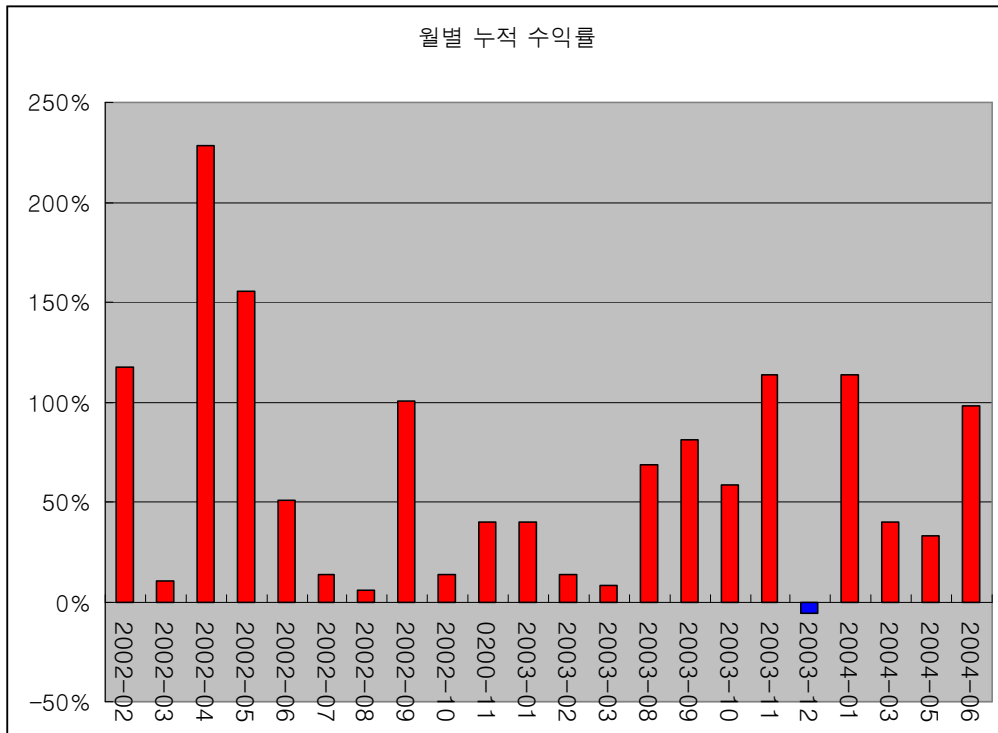
[표 4-4 NShell의 시뮬레이션 결과파일 2]

2) <거래예제2>의 시뮬레이션 결과

시뮬레이션 수행조건	시뮬레이션 수행결과
① 예측치: 0.29이상	수익금: 105,395,092원
② 거래대금 상하한: 5억 -1조	평균보유일수: 3.90일
③ 매수할인율 :1.06	거래당 수익률: 8.11% 일별PPT: 2.08%
④ 이익실현율 구간:1.14	적중률: 71.5% 이익실현 71.5%
⑤ 손절율 구간: 0.82	손절율: 5.2%
⑥ 최대 보유일: 5일	이익: 10.5% 손실: 12.8%

[표 4-5 NNShell의 시뮬레이션 수행 결과 2]

<거래예제2>의 시뮬레이션 수행결과는 <거래예제1>에 비해 낮은 이익실현율 구간을 설정하여 높은 적중률을 보이고 있다. 높은 이익실현율의 경우 전체적인 거래를 통해서인지, 아니면 특정한 달의 높은 수익률 때문인지를 파악할 수 있어야 한다. 월별 표준편차를 사용하여 월별 수익률의 변동을 나타낼 수 있다. <거래예제2>의 월별 표준편차는 약 46으로 매우 안정적인 값을 가지고 있다. 즉, 월별 표준편차가 작으므로 전반적으로 고른 월별 수익률의 분포를 나타내고 있음을 알 수 있다. <거래예제2>는 월별 수익률의 변동이 매우 작음을 나타내고 있다. 높은 월별 수익률을 나타내고 있는 경우가 상대적으로 적어 <거래예제2>의 수익금은 적지만 전반적으로 안정적인 월별 수익률을 나타내고 있다. 전체적으로 고른 월별 수익률의 분포를 나타내고 있으므로 투자자금에 영향을 받지 않을 것으로 보인다. <거래예제2>를 그래프로 나타내어 투자자금에 영향을 받는지를 분석하였다.



[그림4-3 <거래예제2>의 월별 수익률]

<거래예제2>의 경우는 그래프를 통해 알 수 있듯이 투자자금에 영향을 받지 않고 있다. <거래예제2>는 <거래예제1>에 비해 월별 수익률이 고르게 분포하고 있으며, 월별 적중률 또한 약 95%로 매우 높은 적중률을 나타내고 있다. <거래예제1>의 월별 적중률은 74%를 보였고 <거래예제2>의 경우 월별 적중률은 95%로 매우 높으므로 투자자들의 심리에 안정적인 영향을 미칠 것으로 보인다. <거래예제2>의 경우는 월별 적중률이 높고 월별 표준편차가 작으므로 투자자들에게 신뢰감을 준다. 월별 적중률이 높고 월별 표준편차가 작으면 작을수록 안정적인 거래를 이루어질 것으로 판단된다. <거래예제2>는 월별 적중률이 높고 월별 표준편차가 작으므로 이상적인 거래로 보인다.

다른 사례로 월별 적중률이 비슷한 경우의 월별 표준편차를 이용하여 거래의 안정성을 비교하여 투자자들의 심리에 미치는 영향을 파악하고자 한다.

<거래예제 3>

020049-0.400-0.000-0.990-예측치0.400이상 5억-1조 0.98매수 1.24이익실현 0.82손절 9일보유.txt:
 수익금: 376340404원. 보유일수: 6.81일, 일별PPT: 1.47%, 매매회수:578, PPT:9.98%, 이익실현:292(50.5%),
 이익:127(22.0%), 손절:90(15.6%), 손실:69(11.9%), 적중률:50.5%

031330 삼테크# *20010102일 0.83330 20010103일 2400원 20010108일 2976원 23.200 3037370
 037630 벤트리# *20010102일 0.82594 20010103일 16100원 20010105일 19964원 23.200 5952130
 030530 아토# *20010102일 0.77410 20010103일 2280원 20010105일 2827원 23.200 3020628
 041510 에스엠# *20010102일 0.77078 20010103일 5050원 20010104일 6262원 23.200 2701413
 033230 인성정보# *20010102일 0.75106 20010103일 3400원 20010105일 4216원 23.200 7195103
 038390 미디어솔루션# *20010102일 0.73916 20010103일 16200원 20010104일 20088원 23.200 8568600
 023590 다우기술 *20010102일 0.73415 20010103일 2950원 20010108일 3658원 23.200 15056079
 037700 인디시스템# *20010102일 0.73005 20010103일 3510원 20010104일 4352원 23.200 6123083
 036710 심택# *20010102일 0.72181 20010103일 2175원 20010104일 2697원 23.200 2783478
 035830 웰링크# *20010102일 0.70499 20010103일 3500원 20010105일 4340원 23.200 11273337
 032560 황금에스티# *20010102일 0.69092 20010103일 4135원 20010104일 5127원 23.200 4277590
 037340 장미디어# *20010102일 0.67792 20010103일 3100원 20010104일 3844원 23.200 5413214
 036620 버추얼텍# *20010102일 0.67622 20010103일 4260원 20010105일 5282원 23.200 14045992
 038680 에스넷# *20010102일 0.67506 20010103일 6910원 20010108일 8568원 23.200 5356806
 032790 지이티# *20010102일 0.66610 20010103일 855원 20010105일 1060원 23.200 8877458

.....

200101	1185.54%	52번	매매
200103	93.17%	7번	매매
200201	-32.07%	4번	매매
200202	73.63%	11번	매매
200204	297.76%	23번	매매
200205	231.91%	24번	매매
200206	1014.12%	98번	매매
200207	580.06%	45번	매매
200208	4.40%	2번	매매
200209	464.67%	94번	매매
200210	633.25%	47번	매매
200211	114.06%	9번	매매
200212	165.68%	8번	매매
200301	5.43%	5번	매매
200302	1.75%	3번	매매
200303	641.35%	70번	매매
200304	-37.34%	2번	매매
200305	-3.29%	2번	매매
200306	28.46%	3번	매매
200307	33.03%	5번	매매
200308	-84.70%	17번	매매
200309	84.47%	10번	매매
200310	76.74%	6번	매매
200311	110.82%	15번	매매
200312	9.52%	2번	매매
200401	85.26%	12번	매매
200402	-7.76%	2번	매매

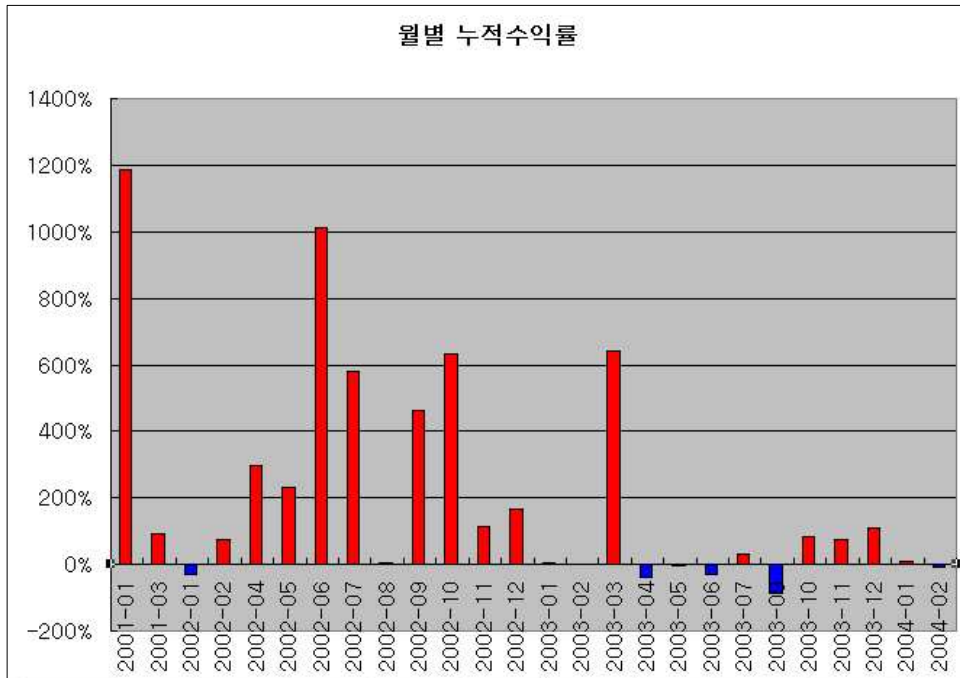
[표 4-6 NNShell의_시뮬레이션 결과파일 3]

3) <거래예제3>의 시물레이션 결과

시물레이션 수행조건	시물레이션 수행결과
예측치: 0.4이상	수익금: 376,340,404원 매매횟수: 578
거래대금 상하한: 5억 -1조	평균보유일수 : 6.81일
매수할인율 :0.98	거래당 수익률 : 9.98 일별PPT:1.47%
이익실현율 구간:1.24	적중률: 50.5% 이익실현율 50.5%
손절율 구간: 0.82	손절율: 15.5%
최대 보유일: 9일	이익: 22.0% 손실: 11.9%

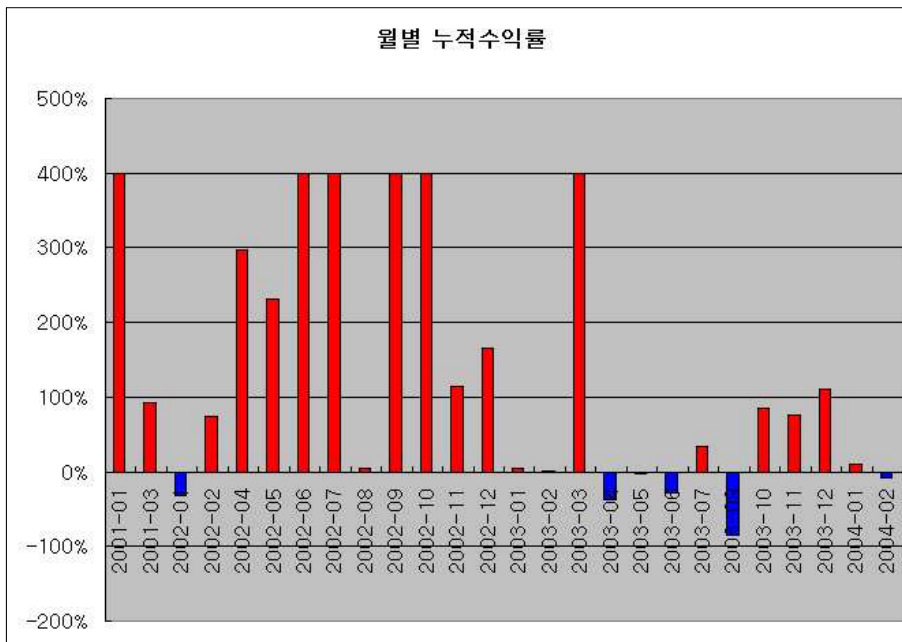
[표 4-7 NShell의 시물레이션 수행결과]

<거래예제3>의 시물레이션 결과를 분석해보면, 예측치 베이스는 0.4 이상, 매수할인율은 0.98로 낮은 가격에 매수하였다. 다른 거래예제들과 비교하였을 거래당 수익률은 비슷하지만 낮은 적중률을 보이고 있다. 전체적인 적중률이 낮은 경우에도 월별 표준편차가 작고 월별 적중률이 높으면 투자자들의 심리에 안정감을 줄 것으로 보인다. 전체적인 적중률이 낮은 경우 적중률이 높은 경우보다 좋지 않은 결과인지를 판단하기 위해 월별 표준편차와 월별 적중률을 이용하여 거래의 안정성을 비교하였다. <거래예제3>의 월별 표준편차는 ≈ 253 으로 나타났고, 월별 적중률은 $\approx 81\%$ 이다. <거래예제3>의 월별 수익률을 [그림 4-4]와 같이 그래프로 나타냈고, 월별 적중률 및 월별 표준편차를 이용하여 다른 거래예제들과 비교 분석하였다.



[그림4-4 <거래예제3>의 월별 수익률]

<거래예제3>은 <거래예제1>과 마찬가지로 자금의 한계로 투자의 제한을 받는다. 그러나 [그림 4-1]과 [그림 4-4]에 나타난 결과를 비교 분석하면, <거래예제3>은 <거래예제1>보다 안정적인 거래가 이루어질 것으로 보인다. 앞서 언급한 바와 마찬가지로 최대 월별 수익률을 400%로 제한하여 [그림 4-5]와 같이 제시하였다.



[그림 4-5 <거래예제3>의 가설 그래프]

<거래예제3>의 경우도 월별 수익률이 최대 한계점에 도달한 경우가 여러 번 나타나고 있으므로 <거래예제1>과 마찬가지로 투자자금에 영향을 받을 것으로 판단된다. 그러나 <거래예제3>은 <거래예제1>보다 월별 표준편차가 작다. <거래예제3>은 월별 수익률의 변동이 <거래예제1>보다 작으므로 비교적 안정적인 거래를 보이고 있다. 즉, <거래예제3>이 <거래예제1>보다는 투자자들의 심리에 안정감을 준다고 말할 수 있다. NNShell의 시뮬레이션 수행결과를 비교 분석하여 새로운 성능 평가 척도로서 월별 표준편차와 월별 적중률을 제안하였다. 앞서 산출한 월별 표준편차와 월별 적중률을 표로 정리하면 다음과 같다.

	월별 표준편차	월별 적중률
<거래예제 1>	376	74%
<거래예제 2>	46	95%
<거래예제 3>	283	81%

[표 4-8 월별 표준편차와 월별 적중률]

<거래예제2>는 월별 표준편차고 작고, 월별 적중률이 매우 높으므로 가장 안정적인 거래 보이고 있으며, 이 결과는 투자자들의 심리에 안정적인 요인으로 작용할 것으로 판단된다. <거래예제1>과 <거래예제3>은 월별 적중률은 큰 차이를 보이고 있지 않지만 월별 표준편차가 작으면 작을수록 투자자들의 심리에 안정감을 준다. 그러므로 <거래예제3>은 <거래예제1>보다는 안정적인 거래를 보인다고 말할 수 있다. 월별 표준편차가 작을수록 전체적으로 고른 수익률의 분포를 나타내고 있기 때문이다.

본 논문에서는 월별 표준편차와 월별 적중률을 이용하여 거래의 안정성을 비교하였다. 월별 적중률이 비슷할 경우 월별 표준편차가 작으면 작을수록 안정적인 거래가 이루어지고 있는 것으로 판단된다. 월별 표준편차와 월별 적중률은 투자자의 심리에 영향을 미치는 새로운 성능 평가 척도로서 유용할 것으로 판단되어진다.

V. 결 론

최근에는 주가를 예측하는 어렵고 복잡한 과정에 과학적이고 효율적인 방법으로 인공지능망 이론 등이 적용되고 있다. 신경망 이론을 적용한 주가예측에 관한 연구사례를 성능 평가의 관점에서 비교 분석하였다. 그 결과 성능 평가 방법으로는 넓게 시계열 접근법과 거래 시뮬레이션 접근법으로 분류되고 있다. 연구사례들의 대부분은 거래 시뮬레이션 접근법을 채택하고 있다. 이 원인을 분석하자면, 주식투자의 최종적인 목표는 RMSE의 최소화가 아니라 수익률의 극대화이므로 시계열 접근법보다 거래 시뮬레이션 접근법이 적합한 것으로 판단된다. 거래 시뮬레이션 접근법이 시계열 접근법보다는 현실적인 요소들을 잘 반영하고 있다. 그러나 대부분의 사례연구들은 주가 예측시스템의 성능을 평가하는 척도로 연간 수익률만을 고려한 매우 단순한 평가 척도들을 채택하고 있다. 단순한 성능 평가 척도만으로는 변동성이 심한 주식시장을 반영하는데 한계가 있을 것으로 판단된다.

이에 본 논문에서는 주식시장의 실제거래상황과 밀접한 관련이 있는 현실적인 요소들을 반영한 “NNShell”의 성능 평가 모형을 분석하였다. NNShell을 분석한 결과는 첫 번째로 예측 파라미터에 해당하는 예측치들의 통계적 유의성을 보다 효과적으로 검토하기 위해 거래 신호의 생성을 위한 다양한 예측 임계치를 고려하고 있다. 두 번째로 매수할인율, 이익실현율, 손절율, 보유기간 등의 다양한 거래정책 인자를 채택하고 있다. 세 번째로 NNShell은 주가예측과 거래정책을 통합한 시뮬레이션 도구로써 효율적인 투자전략 수립이 가능할 것으로 판단된다. 네 번째로 NNShell은 적중률, 거래당 수익률, 수익금등으로 다양한 성능 평가 척도를 채택하고 있다. NNShell이라는 통합 시뮬레이션 도구는 주가예측과 거래정책을 통합하고 있으며 기존 주가 예측시스템보다 다양

한 성능 평가 척도를 채택함으로써 거래 성능을 개선시켰다. NNShell의 성능 평가 척도들의 최적화 의미를 파악하기 위해 실제 데이터를 적용한 실험결과를 예시하였다.

실제 주식시장에서는 투자자금의 한계로 인해 최대 월별 수익률의 한계점이 존재할 것으로 보인다. 이에 본 논문에서는 월별 수익률의 최대 한계점을 400%로 제한하는 가설을 설정하였다. NNShell 펀드 시뮬레이션 결과의 월별 수익률을 그래프로 나타내어 비교 분석하였다. 그 결과 특정한 기간에 집중적인 예측력을 보인 경우가 있다. 이 경우, 이익실현율은 전반적으로 고른 거래 아니라 특정한 기간의 집중적인 수익률 때문인 것을 그래프를 통해 파악할 수 있었다. 이는 주가 예측시스템이 특정한 기간에만 집중적인 예측력을 보인 것으로 단지 주가 예측시스템의 순수한 예측력 때문인지 아니면 그 당시의 주식시장의 호황 때문인지를 판단할 수 있어야 한다. 특정한 기간에만 집중적인 예측력을 보이는 주가 예측시스템은 투자자들에게 신뢰감을 주지 못할 것으로 보인다. 이에 투자자들의 투자 심리를 고려한 거래 시스템의 구성이 요구된다. 이에 본 논문에서는 NNShell에 포함되는 것이 바람직할 것으로 판단된 추가적인 성능 평가 척도로 월별 표준편차와 월별 적중률을 제안 하였다. 월별 표준편차가 크면 클수록 월별 수익률의 변동이 심한 것을 나타내므로 투자자들의 심리에 불안요소로 작용할 것으로 추측된다. 월별 수익률이 고르게 분포해야 안정적인 거래가 성립될 것으로 판단된다. 월별 적중률과 월별 표준편차를 사용하여 NNShell의 시뮬레이션 결과를 비교하였고, 결과는 실질적인 수익금이 감소했다라도 월별 적중률이 높고 월별 표준편차가 작은 경우에 안정적인 거래 성능을 나타냈다.

향후 과제로는 “NNShell 통합 시뮬레이션 도구”의 성능 평가 척도에 월별 표준편차와 월별 적중률을 포함시켜 최적화의 의미를 파악하고, 실제 주식시장에서의 투자자들의 심리에 미치는 영향에 대한 연구가 이루어져야 한다.

참고 문헌 및 웹 사이트

- [1] 이장우, 강인철 “증권시장론” 대명 2003.
- [2] T. Hellstrom “A Random walk through the stock market” ph *D.* Thesis. Department of Computing Science. Umea University. 1998.
- [3] T. Hellstrom and K. Homstrom, Prediction the Stock Market, Technical Report IMA-TOM-1997-07, Department of Mathematics and Physics, Malardalen University, Sweden, 1997.
- [4] T. Hellstrom “Data Snooping in the Stock Market” 1998.
- [5] Journana Ghosn, Yoshua Bengio, “Multi Task Learning for Stock Selection”pp 947-952 1998.
- [6] Tim chenoweth, Zoran obradovic, Sauchi Stephen Lee “Neural Network Trading Systems” 1996.
- [7] James *D* Thomas, Katia Sycara “Integrating Genetic Algorithms and Text Learning for Financial Prediction” 2000.
- [8] “증권시장지표 해설” 한국증권 거래소 2000.
- [9] “현대 증권”, <http://www.youfirst.co.kr> .
- [10] J. O. J W Lee, and B chae, “Dynamic Asset Allocation Exploitation Predictor in Reinforcement Framework”, European Conference on

Machine Learning, pp 298–309.

- [11] Jangmin O, Jae Won Lee, Jongwoo Lee “*Dynamic Asset Allocation Exploiting Predicators in Reinforcement Learning Framework*” 2004.
- [12] J.W Lee, S. Kim, J. Lee, and J. chae, “An Intelligent stock Trading System based on Reinforcement Learning”, *IEICE Trans on Information and System* 2003.

ABSTRACT

A Case study of Performance Evaluation of Stock Time Series Prediction based on Machine Learning

Moon, Jung hee

Major in Computer Science Education

Graduate School of Education

Sungshin Women's University

It is difficult to predict the change of stock price series or the expected profit because of non-stationarity. In general, to predict stock prices, there are fundamental analysis which estimates inherent value of the company and technical analysis which uses past price patterns. Lately, the studies on predicting stock prices based on neural networks showed some promising results in more scientific and efficient ways. Although predicting stock market using neural networks have been consistently done, we still cannot be sure whether it can give satisfying results to the investors in the real stock market.

In addition, we present additional and various metrics for performance evaluation considering more realistic considerations, using an integrated simulation tool called "NNShell", introduce the concept of "integrated simulation" which combines prediction result with trading strategy, and also propose additional metrics to be incorporated with NNShell for the purpose of enhancement of that tool.

기계학습
기반의
주가시계열
예측에서의
성능
평가
연구사례

문
정
희