The Journal of the Convergence on Culture Technology (JCCT)
Vol. 9, No. 2, pp.475-481, March 31, 2023. pISSN 2384-0358, eISSN 2384-0366

http://dx.doi.org/10.17703/JCCT.2023.9.2.475

JCCT 2023-3-59

머신러닝 애플리케이션 구현 비용 평가를 위한 확장형 기능 포인트 모델

An Extended Function Point Model for Estimating the Implementing Cost of Machine Learning Applications

임석진*

Seokjin Im*

요 약 머신러닝과 같은 소프트웨어가 일상생활에 매우 큰 영향력을 발휘하고 있는 상황에서, 소프트웨어의 개발비용을 평가하는 비용 모델의 중요성이 지속적으로 증가하고 있다. 비용 모델로서 LOC(Line of Code)와 M/M(Man-Month) 모델은 소프트웨어의 양적인 요소들을 측정하는 비용모델이다. 이와는 달리, FP(Function Point)는 소프트웨어의 기능적 특징들을 평가하는 비용모델로서 소프트웨어의 질적인 요소를 평가한다는 점에서 효과적이다. 그러나 FP는 머신러닝 소프트웨어의 주요한 요소들을 평가하지 않기 때문에 머신러닝 소프트웨어를 평가하는데 한계를 가진다. 본 논문은 확장형 FP(Extended Function Point, ExFP)를 제안한다. 확장형 FP는 머신러닝의 주요 특징인 하이퍼 파라미터와 그것의 최적화에 대한 복잡도를 반영하여 소프트웨어의 기능적 요소를 평가하도록 확장하였기 때문에 머신러닝과 같은 최신 소프트웨어에의 비용 평가에 적합하다. 머신러닝 소프트웨어의 특징을 반영한 평가를 통해 제안된 확장형 FP의 효용성을 보였다.

주요어 : 비용모델, 기능 포인트, 머신러닝

Abstract Softwares, especially like machine learning applications, affect human's life style tremendously. Accordingly, the importance of the cost model for softwares increases rapidly. As cost models, LOC(Line of Code) and M/M(Man-Month) estimates the quantitative aspects of the software. Differently from them, FP(Function Point) focuses on estimating the functional characteristics of software. FP is efficient in the aspect that it estimates qualitative characteristics. FP, however, has a limit for evaluating machine learning softwares because FP does not evaluate the critical factors of machine learning software. In this paper, we propose an extended function point(ExFP) that extends FP to adopt hyper parameter and the complexity of its optimization as the characteristics of the machine learning applications. In the evaluation reflecting the characteristics of machine learning applications. we reveals the effectiveness of the proposed ExFP.

Key words : Cost Model, Function Point, Machine Learning

*정회원, 성결대학교, 컴퓨터공학과 조교수 (제1저자)
접수일: 2022년 12월 30일, 수정완료일: 2023년 2월 23일
게재확정일: 2023년 3월 8일

Received: December 30, 2022 / Revised: February 23, 2023
Accepted: March 8, 2023
*Corresponding Author: imseokjin@sungkyul.ac.k
Dept. of Computer Engineering, Sungkyul Univ, Korea

Ⅰ. 서 론

소프트웨어 애플리케이션은 고속 네트워크와 하드웨어의 발전 뿐만 아니라 다양한 컨텍스트를 처리하는 소프트웨어 기술 발전에 힘입어 엄청난 속도와 규모로 인간의 라이프 스타일에 영향을 미치고 있다[1, 2]. 예를 들어 손목의 스마트 워치는 심박수를 24시간 측정하여 건강 상태를 모니터링하여 알람을 보냄으로서 생활의 질적인 변화를 이끌고 있다. 특히 머신러닝(Machine Learning) 애플리케이션은 생활의 편리함을 더욱 높여주고 있고 적용되는 범위가 확대되어 중요성이 점점 더 높아지고 있다. 머신러닝 애플리케이션의 경우 소프트웨어의 복잡성이 지속적으로 증가하고 있기 때문에 소프트웨어 개발비용을 추정하는 모델의 중요성이 더욱 확대되고 있다 [3].

비용 모델은 소스 코드의 라인 수와 같은 소프트웨어의 정량적 측면 또는 애플리케이션 기능의 복잡성과 같은 정성적 측면을 측정하여 소프트웨어의 개발비용 또는 유지보수 비용을 평가한다. LOC(Line of Code)는 응용 프로그램의 소스 코드의 라인 수로 소프트웨어 비용을 평가하는 비용모델이다. LOC는 단순하여 평가하기가 쉽지만 소스 코드에 주석이 많이 포함된 경우 비용을 과대 평가할 수 있는 문제가 있다. M/M(Man-Month)은 개발자 수와 예상 개발 기간을 사용하여 소프트웨어 비용을 평가하는 모델이다. M/M은 경험적이고 주관적인 관점으로 비용을 평가할 수 있는 문제를 피할 수 없다. 이와같이, LOC 및 M/M은 소프트웨어의 질적 측면이 아닌 양적 측면에 의존하는 모델이라는 한계를 가진다.

FP(Function Point)는 LOC와 M/M의 대안으로 제안된 비용모델로서, 소프트웨어의 기능과 같은 질적인 측면을 고려하여 개발비용을 평가한다 [4, 5]. FP는 데이터 기능(Data Function)과 트랜잭션 기능(Transaction Function)을 평가하며, 데이터 기능은 애플리케이션의 데이터 처리 복잡도를 평가하고, 트랜잭션 기능은 파일 읽기나 쓰기, 또는 데이터베이스 트랜잭션 처리와 같은 기능의 복잡도를 평가한다. 이와같이 FP는 파일 처리 및 데이터베이스 처리의 복잡성과 데이터 처리 기능의 복잡도에 집중하기 때문에 트랜잭션 및 파일을 처리하는 소프트웨어 비용평가에 적합하다.

UCP(Use Case Point)는 유스케이스 단위로 애플리케이션을 개발하는 객체지향 기반 소프트웨어 비용 평가를 위해 개발된 비용평가 모델이다 [6]. UCP는 소프트웨어에서 식별된 유스케이스에 대해 개발비용을 평가하기 때문에 객체 지향 패러다임의 소프트웨어 비용 평가에 적합하다. FP와 UCP는 애플리케이션의 질적 측면을 이용하여 개발비용을 평가하기 때문에 양적인 측면에 초점이 맞춰진 LOC 또는 M/M 보다 합리적인 비용 모델이다. 그러나 FP와 UCP는 개발과정에서 데이터 전처리와 학습과정에서 소프트웨어 파라미터에 대한 복잡한 최적화를 수반하는 머신러닝 소프트웨어의 특수성을 고려할 수 없기 때문에 머신러닝 기반 소프트웨어의 개발비용을 평가하기에는 부족한 문제를 가지고 있다.

본 논문에서는 기존의 FP 비용 모델이 갖고 있는 문제를 해결하기 위해 데이터 전처리의 복잡도와 학습 과정의 파라미터 최적화의 복잡도를 고려해서 머신러닝 애플리케이션 개발비용을 평가할 수 있는 확장형 FP(ExFP, Extended Function Point) 모델을 제안한다.

Ⅱ장에서는 소프트웨어 개발비용을 평가하는 비용 모델로서의 FP의 세부 내용을 리뷰하고, Ⅲ장에서 머신러닝 애플리케이션의 개발비용을 효과적으로 평가하기 위한 ExFP 비용 모델을 제안한다. 다음으로 Ⅳ장에서 ExFP와 FP를 비교하여 제안 모델의 유효성을 보이고, 마지막으로 Ⅴ장에서 논문의 결론을 맺는다.

Ⅱ. 기능 포인트(FP) 모델

소프트웨어 시스템의 기능 포인트 평가는 아래 그림 1에 보인 절차를 따르며 다음과 같이 5단계로 구성된다.

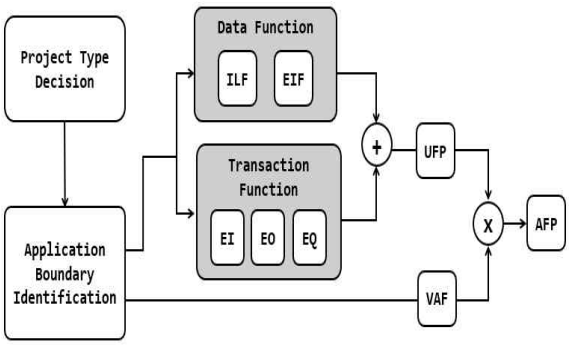

그림 1. 기능 포인트 평가를 위한 절차
Figure 1. Procedure for estimating function points

1. 프로젝트 유형 결정

소프트웨어 애플리케이션의 기능 포인트를 측정하기 위해 먼저 프로젝트 유형을 개발 프로젝트, 개선 프로젝트 및 개발 완료된 응용 프로그램 중 하나로 결정한다. 이는 프로젝트 유형에 따라 평가 대상 소프트웨어의 비용을 평가하기 위한 기능 포인트 측정 범위가 달라지기 때문이다.

2. 애플리케이션 경계 식별

이 단계는 애플리케이션 기능의 측정 범위를 결정하기 위한 애플리케이션의 경계를 식별한다. 개발 프로젝트의 경우 프로젝트에서 제공해야 하는 모든 기능이 측정 범위에 포함되어야 한다. 이 단계에서 기능 포인트를 측정하기 위해 개발되어야 하는 모든 논리적 기능을 식별한다 [7].

3. UFP 측정

UFP(Unadjusted Function Point)는 아래의 식 (1)을 이용해 측정하며, UFP는 데이터 기능 포인트와 트랜잭션 기능 포인트의 합이다.

$$UFP = \Sigma(V_{ILF} , V_{EIF} , V_{EI} , V_{EO} , V_{EQ}) \quad (1)$$

1) 데이터 기능 포인트 측정: 이를 위해 데이터 기능에 대한 ILE 및 EIF의 두 가지 요소를 식별하고 측정한다. ILE(Internal Logical Files)는 식별 가능하고 논리적으로 관련된 데이터 또는 제어 정보 그룹에 대한 복잡도로서 소프트웨어 시스템에서 관련된 파일 수로 측정된다. ELF(External Interface File)는 응용 시스템이 참조하는 파일에 대한 복잡도로서, ILF처럼 파일 수로 측정한다. EIF와 ILF는 표 1에 보인 것처럼 DET(Data Element Type)와 RET(Record Element Type)를 이용하여 측정된다 [6]. 표 1에서 DET는 ILE 및 EIF의 고유 필드 수를 의미하고 RET는 DET의 하위 집합인 레코드 수를 의미한다.

표 1. ILE와 ELF 평가를 위한 매트릭스
Table 1. Matix for ILF and ELF

		DET		
		1 to 19	20 to 50	over 51
RET	1	Low	Low	Average
	2 to 5	Low	Average	High
	over 6	Average	High	High

2) 트랙잭션 기능 포인트 측정: 트랜잭션 기능 포인트로서 EI, EO 및 EQ의 세 가지 요소를 식별하고 측정한다.

EI(External Input)는 응용 시스템 외부에서 시스템으로 요청되는 등록, 수정, 삭제와 같은 제어 정보와 같은 입력 데이터 또는 단위 프로세스에 대한 복잡도를 의미하며 참조 테이블인 DET 및 FTR(File Type Reference)로 측정된다. 아래의 표 2에 보인 DET 및 FTR 값을 이용하여 EI를 측정한다 [6].

표 2. EI 평가를 위한 매트릭스
Table 2. Matix for EI

		DET		
		1 to 4	5 to 15	over 16
FTR	0 to 1	Low	Low	Average
	2	Low	Average	High
	over 3	Average	High	High

EO(External Output)는 데이터 처리 후 소프트웨어 시스템에서 외부로 나가는 출력 데이터 또는 제어 단위 프로세스에 대한 복잡도를 의미한다. EQ(External Query)는 소프트웨어 외부와 소프트웨어 시스템 간의 입출력 데이터 질의에 대한 복잡도를 의미한다. EO와 EQ도 EI와 같이 DET와 FTR를 이용하여 측정되며, 아래의 표 3에 표시된 것과 같다 [6].

표 3. EO와 EQ 평가를 위한 매트릭스
Table 3. Matix for EO and EQ

		DET		
		1 to 5	6 to 19	over 20
FTR	1	Low	Low	Average
	2 to 3	Low	Average	High
	over 4	Average	High	High

4. VAF 측정

VAF(Value Adjustment Factor)는 소프트웨어 시스템의 일반적인 기능을 측정하는 요소이다. VAF는 아래의 표 4에 보인 것과 같이 14개 항목을 포함하는 일반 시스템 특성(GSC, General System Characteristics)로 측정한다. 표 4의 각 항목은 아래의 표 5에 보인 것처럼 0에서 5까지의 영향도(DI, Degree of Influence) 값을 갖는다.

GSC의 항목에 대한 DI 값을 사용하여 VAF는 다음의 식 (2)와 같이 계산된다.

$$VAF = 0.01\Sigma(DI_i) + 0.65 \quad (2)$$

표 4. 일반 시스템 특성
Table 4. General System Characteristics

GSC		Value
G_1	Data Communicatiaon	DI_1
G_2	Distributed Data Processing	DI_2
G_3	Performance	DI_3
G_4	Heavily Used Configuration	DI_4
G_5	Transaction Rate	DI_5
G_6	Online Data Entry	DI_6
G_7	End-User Efficiency	DI_7
G_8	Online Update	DI_8
G_9	Complex Processing	DI_9
G_{10}	Reusablilty	DI_{10}
G_{11}	Installation Ease	DI_{11}
G_{12}	Operation Ease	DI_{12}
G_{13}	Multiple Sites	DI_{13}
G_{14}	Facilitate Change	DI_{14}

표 5. 영향도
Table 5. Degree of Influence

Value	Meaning
0	No influence
1	Incidential influence
2	Moderate influence
3	Average influence
4	Significant influence
5	Strong influence throughout

5. AFP 계산

AFP(Adjusted Function Point)는 소프트웨어 시스템의 전체 기능 점수로서, VAF에 의해서 조정된 UFP를 의미한다. UFP와 VAF를 사용하여 AFP는 다음의 식 (3)을 이용하여 계산할 수 있다.

$$AFP = UFP \cdot VAF \quad (3)$$

III. 확장형 기능 포인트 모델 (Extended Function Point Model)

1. 머신러닝 애플리케이션 개발 절차

머신러닝 애플리케이션 개발은 그림 2와 같이 데이터 수집, 머신러닝 모델 설계 및 구현, 그리고 모델 테스트 단계를 따른다.

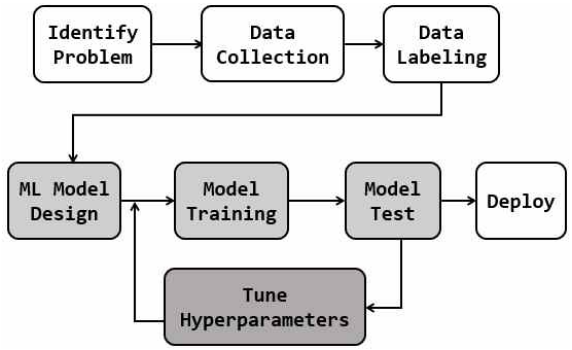

그림 2. 머신러닝 애플리케이션 개발 절차
Figure 2. Procedure for developing ML application

1) 문제 정의: 첫 번째 단계로 개발하려고하는 머신러닝 애플리케이션이 해결해야 할 문제가 분류, 회귀, 추천 등과 같은 머신러닝 문제 중에서 어떤 문제인지를 결정한다.

2) 데이터 수집과 레이블링: 정형 데이터, 비정형 데이터, 정적 데이터 또는 스트리밍 데이터와 같이 머신러닝 애플리케이션에서 사용할 데이터 종류와 필요한 데이터의 양을 결정한다. 그런 다음 데이터를 수집하고 누락된 값을 처리하거나 데이터 필터링과 같은 전처리를 수행한다. 수집된 데이터는 레이블링을 통해 훈련 데이터와 테스트 데이터, 두 그룹으로 나눈다. 이 단계는 머신러닝 애플리케이션의 신뢰도를 높이는데 결정적인 역할을 한다.

3) 머신러닝 모델 설계와 학습: 정의된 문제를 해결할 수 있는 머신러닝 모델의 유형과 학습에 사용할 알고리즘을 결정하여 머신러닝 모델을 설계한다. 또한 모델의 특징과 성능에 관련된 중요 하이퍼 파라미터를 추출하고 초기화한다. 특히 하이퍼 파라미터는 머신러닝 애플리케이션의 성능과 정확도, 그리고 학습하는 과정의 난이도에 영향을 미치는 결정적 요소이다. 이후, 학습 데이터와 초기화된 파라미터를 사용하여 모델을 학습하고 검증을 위해 테스트 데이터로 학습된 머신러닝 모델을 테스트한다.

4) 하이퍼 파라미터 조정: 이 단계는 정확도와 성능을 보장하기 위해 머신러닝 모델의 하이퍼 파라미터를 최적화하는 단계로 하이퍼 파라미터 조정은 머신러닝 모델의 성능에 매우 중요한 역할을 한다. 하이퍼 파라미터의 값을 조정하고, 조정된 파라미터 값을 사용하여 모델을 재학습한 후 정확성과 성능을 테스트한다. 정확

도와 성능이 목표값에 도달할 때까지 이 단계를 반복적으로 실행한다.

5) 애플리케이션 배포: 최적화 과정이 끝난 후, 개발된 머신러닝 애플리케이션을 배포한다.

2. 확장형 기능 포인트 모델

머신러닝 애플리케이션에서 모델 훈련을 위한 데이터 준비 및 모델의 하이퍼 파라미터 추출 및 최적화는 모델의 성능 및 정확도 뿐만 아니라 개발에 소요되는 시간을 결정하는 중요한 요소이다. 따라서 머신러닝 애플리케이션 개발을 위한 비용 모델에는 데이터 준비 및 하이퍼 파라미터, 이 두 가지 중요한 요소가 포함되어 이 두 가지 요소에 따른 개발비용의 영향을 고려해야 한다. 여기서 FP를 확장하는 ExFP(Extended Function Point)를 구성하여, 머신러닝 애플리케이션에서 데이터 준비 및 하이퍼 파라미터의 영향을 비용 모델에 반영하여 머신러닝 애플리케이션에 실효성있는 비용모델을 제안한다.

1) 하이퍼 파라미터 복잡도

제안된 ExFP는 비용 모델에 하이퍼 파라미터의 영향을 반영하기 위해 하이퍼 파라미터의 복잡도인 HPR을 적용한다. HPR은 하이퍼 파라미터를 최적화하는 과정의 복잡도를 의미한다. HPR은 데이터를 처리하는 트랜잭션과 관련이 있기 때문에 트랜잭션 기능 중 하나로 처리되어야 하며, HPR의 값은 FP의 다른 트랜잭션 기능 포인트와 마찬가지로 Low, Average 및 High 중 하나의 값을 갖는다. ExFP의 UFP는 다음과 같이 식 (4)를 이용하여 측정되며, UFP 계산에 사용되는 각 항목의 복잡도는 표 6과 같다.

$$UFP = \Sigma(V_{ILF}, V_{EIF}, V_{EI}, V_{EO}, V_{EQ}, V_{HPR}) \quad (4)$$
$$= N_{UF} * ILF + N_{EI} * EIF + N_{UI} * EI + N_{UO} * EO + N_{UQ} * EQ + N_{HP} * HPR$$

여기서, N_{UF}는 사용자 파일의 수이며, N_{EI}는 외부 인터페이스의 수이다. N_{UI}는 사용자 입력의 수이고, N_{UO}는 사용자 출력의 수이다. N_{UQ}는 사용자 질의 수이며, N_{HP}는 머신러닝 모델에서 사용되는 하이퍼 파라미터 수이다.

표 6. ExFP의 UFP를 측정을 위한 파라미터 복잡도
Table 6. Complexity of parameters for UFP of ExFP

	Complexity		
	Low	Average	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
HPR	5	7	10
ILF	7	10	15
EIF	5	7	15

2) 확장형 일반 시스템 특성 (Extended General System Characteristics, ExGSC)

제안된 ExFP는 VAF를 측정하기 위해서 데이터 준비 및 하이퍼 파라미터 최적화의 난이도를 의미하는 항목을 일반 시스템 특성에 추가한다. 추가된 항목 DI_{DP}는 데이터 준비(Data Preparation)에 대한 영향도를 반영하는 항목이고, DI_{PO}는 파라미터 최적화에 대한 영향도를 반영하는 추가된 항목이다. ExFP에서는 두 가지 항목이 추가되어 16개 항목을 포함하는 확장된 GSC (ExGSC) 표를 사용한다.

ExGSC를 이용하여 ExFP의 VAF는 다음과 같이 식 (5)를 이용하여 측정된다.

$$VAF = 0.01\{ \Sigma DI_i + DI_{DP} + DI_{PO} \} + 0.65 \quad (5)$$

여기서 ΣDI_i는 FP의 GSC의 14개 항목을 이용한 값의 합을 의미한다.

ExFP에 대한 AFP는 식 (3)을 이용하여 FP와 같은 방식으로 측정된다.

Ⅳ. 확장형 기능 포인트 모델 평가

머신러닝 애플리케이션에 대해 제안된 ExFP의 개발 비용 평가의 적합성과 효율성을 보인다. 제안된 ExFP 비용 모델의 머신러닝 애플리케이션에 대한 적합도를 평가를 위해서 ExFP와 FP의 UFP, AFP 및 AFP를 이용해 애플리케이션 개발에 소요되는 예상 시간을 추정하여 비교한다. 표 6, 7 및 8은 평가에 사용된 항목들의 값을 보인다. 표 6의 HPR, 표 7의 N_{HP}, 그리고 표 8의 G_{15}와 G_{16}은 ExFP에서 적용한 머신러닝 애플리케이션의 특성을 반영하는 항목이다.

표 7. 평가를 위한 시스템 요소 값
Table 7. values of system factors for evaluation

	Meaning	Value
N_{UI}	the number of user inputs	45
N_{UO}	the number of user outputs	25
N_{UQ}	the number of user queries	43
N_{UF}	the number of user files	20
N_{EI}	the number of external interfaces	10
N_{HP}	the number of hyper parameters	25

표 8. EI 평가를 위한 영향도와 일반시스템 특성 값
Table 8. DI values of GSC for the evaluation

GSC		Value
G_1	Data Communicatiaon	2
G_2	Distributed Data Processing	5
G_3	Performance	5
G_4	Heavily Used Configuration	5
G_5	Transaction Rate	3
G_6	Online Data Entry	3
G_7	End-User Efficiency	4
G_8	Online Update	2
G_9	Complex Processing	5
G_{10}	Reusablilty	3
G_{11}	Installation Ease	3
G_{12}	Operation Ease	4
G_{13}	Multiple Sites	2
G_{14}	Facilitate Change	4
G_{15}	Data Preparation	5
G_{16}	Optimization Parameters	5

1) ExFP와 FP의 UFP 비교

ExFP의 UFP는 표 6, 7, 8에 주어진 항목 값으로 식 (4)를 이용하여 측정한다. 그림 3에서 레이블 FP는 FP의 UFP를 의미하고, ExFP(L)은 복잡도 HPF가 Low일 때 ExFP의 UFP를 의미한다. 이와같이 그림 3에서 ExFP(A)와 ExFP(H)는 복잡도 HPF의 값이 Average일 때와 High일 때 ExFP의 UFP를 의미한다. 그림 3은 ExFP가 머신러닝 애플리케이션에 더 효과적이고 적합한 비용 모델임을 보여준다. 이는 FP의 UFP는 머신러닝 애플리케이션에서 중요한 요소인 하이퍼 파라미터의 개수와 복잡도에 따라 영향을 받지 않기 때문이다. FP와 달리 ExFP에 의한 UFP는 복잡도 HPR이 증가함에 따라 증가하여 머신러닝 애플리케이션의 구현 복잡도에 따라 영향을 받는 것을 보인다.

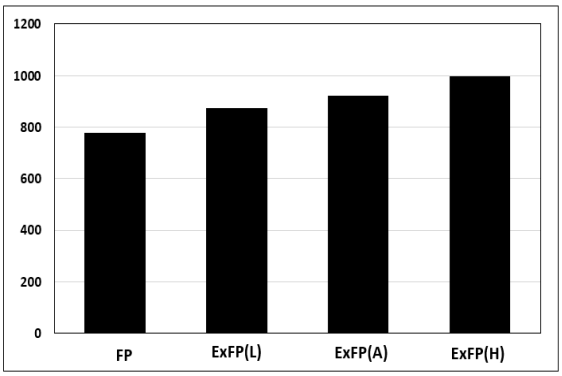

그림 3. 제안된 ExFP와 FP의 UFP
Figure 3. UFP of FP and proposed ExFP

2) ExFP와 FP의 AFP 비교

AFP는 UFP와 VAF가 적용된 애플리케이션 개발비용을 의미한다. 그림 4는 FP와 ExFP에 의한 AFP를 보인다. AFP의 값을 그림 3에서 UFP의 값과 비교하면 AFP의 값이 더 크며, 이것은 AFP가 식 (3)에 의해서 UFP에 VAF 값이 적용된 값이기 때문이다. 그림 4에서 FP의 AFP와 ExFP의 AFP 값의 편차가 그림 3의 UFP 사이의 편차보다 크며, 이는 ExFP에 의해 측정된 VAF 값이 FP에 의한 값보다 크기 때문이다. ExFP의 VAF에는 데이터 준비 및 하이퍼 파라미터 최적화, 두 항목이 추가되어 적용되었기 때문에 ExFP가 머신러닝 애플리케이션 개발비용 측정에 적합한 것을 나타낸다.

3) AFP를 이용해 추정된 개발 시간 비교

식 $T_{EST} = AFP * F_{EST}$을 이용하여 머신러닝 애플리케이션을 구현하는 데 소요되는 시간을 추정할 수 있다. 식에서 T_{EST} 는 개발에 소요될 것으로 예상 시간을 의미하며 F_{EST} 기능 점수당 예상 소요 시간을 의미한다. 그림 5는 F_{EST}의 값이 2일 때 예상 시간을 보인다. 그림 5는 ExFP에 의한 AFP 기반 추정 시간이 FP에 의한 것보다 크다는 것을 보이며, 이는 ExFP가 FP보다 머신러닝 애플리케이션의 특성을 더 효과적으로 반영하기 때문이다.

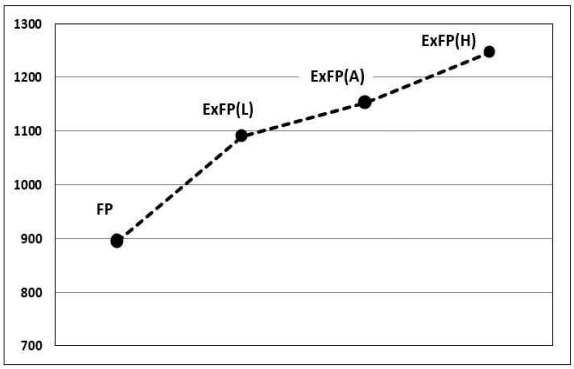

그림 4. 제안된 ExFP와 FP의 AFP
Figure 4. AFP of FP and proposed ExFP

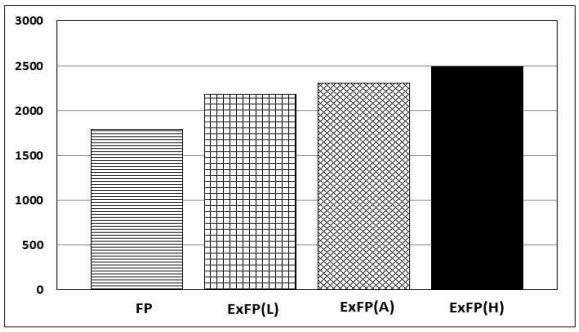

그림 5. AFP를 이용해 추정된 개발 시간
Figure 5. Development time estimated from AFP

V. 결론

본 논문에서 FP를 확장한 확장형 FP (ExFP)를 통해 머신러닝 애플리케이션의 특성을 반영한 비용모델을 제안하였다. ExFP는 비용모델의 UFP가 머신러닝 애플리케이션에 적합하도록 하기 위해서 하이퍼 파라미터의 복잡도를 도입하였다. 또한 ExFP의 ExGSC에는 훈련에 필요한 데이터 준비 과정의 영향 정도와 하이퍼 파라미터 최적화의 영향 정도를 의미하는 두 가지 요소를 추가하여 소프트웨어 개발비용 평가가 머신러닝 애플리케이션의 적합하도록 하였다. 제안된 ExFP가 머신러닝 애플리케애션에 적합합을 평가하기 위해 ExFP와 FP의 UFP를 비교하였고, AFP 및 AFP를 이용하여 개발 소요 시간 추정 분석하여 비교하였다. 평가를 통해 제안된 ExFP의 머신러닝 애플리케이션의 개발비용을 추정하는 데 적합하다는 것을 보였다.

References

[1] D. Seo, and H. Lim, “Convergence Plan of IT Social Safety and SIB by Expanding Sharing Information Data,” *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 8, No. 6, pp. 97-105, November 2022. DOI:http://dx.doi.org/10.17703/JCCT.2022.8.6.97

[2] S. Oh, and K. Han, “Effect of Education about Blockchain Technology on Trust, Security, and Technology Acceptance Model of Virtual Assets,” *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 8, No. 6, pp. 675-683, November 2022. DOI: http://dx.doi.org/10.17703/JCCT.2022.8.6.675

[3] Y. Lee, “Modeling of AutoML using Colored Petri Net,” *The International Journal of Advanced Culture Technology (IJACT)*, Vol. 10, No. 4, pp. 420-426, December 2022. DOI: https://doi.org/10.17703/IJACT.2022.10.4.420

[4] L. Lavazza, and R. Meli, “An Evaluation of Simple Function Point as a Replacement of IFPUG Function Point” IEEE Proceeding of ICSPPM 2014. DOI:10.1109/IWSM.Mensura.2014.28

[5] S. Patel, “Function Point distribution using maximum entropy principle” IEEE Proceeding of ICIIP 2013. DOI: 10.1109/ICIIP.2013.6707682

[6] J. Park, and K. Chong “A UCP-based Model to Estimate the Software Development Cost” The Journal of Korea Information Processing Society D, Vol. 11, No. 1, February 2004.

[7] S. Patel, “Function Point distribution using maximum entropy principle” IEEE Proceeding of ICIIP 2013. DOI: 10.1109/ICIIP.2013.6707682