



YAPAY SİNİR AĞLARI TABANLI YAZILIM EFOR TAHMİNİ

Muaz GÜLTEKİN

Oya KALIPSIZ

Özet

Yazılım efor tahmini bir yazılım mühendisliği projesini geliştirmek için gerekli olan her türlü kaynağın önceden tahmin edilmesi işlemidir. Yazılım efor tahmini kavram olarak basit olsa da, gerçekte zor ve karmaşıktır. Bu yüzden birçok yazılım projesi öngörülen zamanda bitirilememiş ya da proje masrafları düşünülen miktardan çok fazla olmuştur. Yazılım projelerinin farklı aşamalarında yapılan bütçe çalışmalarında, maliyet tahmin zorlukları dolayısıyla, satış tutarı ve maliyet analizleri gerçekçi hesaplanamamaktadır. Bu zorluklar, projenin kendine özgü özelliklerinden kaynaklandığı gibi, kontrol dışı bilgi eksikliğinden, bilgilerin değerlendirilmesindeki öznellikten kaynaklıdır. Maliyet analiz çalışmalarında, direkt ve indirekt maliyet ayırım hatalarından ve proje risklerinin tam olarak doğru tahmin edilememesinden kaynaklanabilmektedir. Yazılım projesine başlarken projenin ne kadar sürede biteceği, proje maliyeti, projede çalışacak kişi sayısı gibi birçok etmen önceden tahmin edilerek proje oluşturulmalıdır. Yazılım projelerinin kaynak tahmininin doğruluğu ve güvenilirliği yazılım projesinin gidişatı için çok önemlidir. Yazılım geliştirme teknolojisinin sürekli değişen senaryolar içinde olması efor tahmini daha zorlu hale getirmektedir. Yazılım projelerinin efor tahmininin doğruluğu ve güvenilirliği yazılım şirketlerinin rekabeti açısından önemlidir. İyi tahminler yazılım projeleri yönetiminde çok önemli bir rol oynamaktadır. Yazılımın etkili ve verimli gelişimi doğru tahminler gerektirir. Bu çalışmamızda geliştirdiğimiz model ile daha gerçekçi yazılım maliyet tahmini yapmaya çalıştık. Yazılım projesi maliyet tahmini için önerdiğimiz model COCOMO ve Yapay sinir ağları tabanlıdır. Geliştirdiğimiz bu model ile yazılım projelerinin efor tahminini yapay sinir ağları yöntemiyle önceden tahmin ederek yazılımın maliyeti, süresi ile ilgili bir takım politikaların önceden belirlenmesine katkı sağlamaktadır.

Anahtar Kelimeler: Yazılım Mühendisliği, Yazılım Maliyet Tahmini, Cocomo, Yapay Sinir Ağları

SOFTWARE EFFORT ESTIMATION USING ARTIFICIAL NEURAL NETWORK

Abstract

Software effort estimation is the process of estimating the resources required to develop various kinds of software project in advance. Software effort estimation, although simple in concept, is practically difficult and complex process. So many software projects could not be completed in the estimated time or the cost of the project has been much more than the amount that is considered. Planning the budget for each stage of software development life cycle is difficult due to software development phase characteristic. So that brings more uncertainty in computation of the total budget estimation such as sales amount and analysis effort. The distinctive features of the project as a result of these challenges, out of control, lack of knowledge, information subjective comments in the evaluation of cost analysis, in distinction from the mistakes of the direct and indirect costs, and project risks can be caused by an inability fully accurate estimated. Before the start the project there are many factors must be defined and estimated correctly such as time, cost, scope and the number of the people. The accuracy of the estimate effort of software projects and software reliability is very important to the progress of the project. The estimation of software development effort to be constantly changing technology makes it more demanding scenarios. The accuracy of the estimate effort of software projects is very important for the competitiveness of software companies and reliability. Good predictions plays a very important role in the management of software projects. The effective and efficient development of software requires accurate estimates of. In this study

we develop software effort estimation model which try to serve more accurate estimation. The suggested software effort estimation model is based on Constructive Cost Model (COCOMO) and Artificial Neural network (ANN). Thanks to suggested model we can predict software effort which involve of time and required number of the people. So Software Company can constitute their software development policy via this model

Keywords: Software Engineering, Software Cost Estimation, Cocomo , Artificial Neural Network

1. GİRİŞ

Bir yazılımın maliyet öngörüsünü geliştirme süreci herhangi başka bir şeyin maliyet öngörüsünü geliştirme sürecinden farklı değildir. Ama bir yazılımın maliyetinin tahmin edilmesinde kendisine has bazı yönleri vardır. Yazılım maliyeti hesaplamasının kendine özel yönlerinin nedeni yazılımın doğası gereği üretilmiş bir ürün olmasıdır. Diğer problemler hesaplama metodlarının doğasından kaynaklanmaktadır [1]. Yazılım maliyet hesaplama süreci, ilk teklif aşamasıyla başlayan ve projenin kaldırılmasına (lift time ı böyle çevirdim) kadar devam eden bir süreçtir. Maliyet hesaplama süreci, harcama ve bütçenin aynı çizgide gitmesinden emin olmaktır.

Maliyet hesaplama, proje yönetiminin en zorlayıcı görevlerinden birisidir. İhtiyaç duyulan kaynakların ve gerekli yazılım geliştirme projelerinin takviminin doğru şekilde hesaplanmasıdır. Yazılım maliyet hesaplama süreci; üretilecek yazılımın boyutunun hesaplanması, bunun için gerekli olan eforun hesaplanması, proje zamanlama planının geliştirilmesi ve son olarak ta projenin tamamının maliyetinin hesaplanmasını içerir [2].

Dış kaynaklı projelerde fiyatlandırma en önemli parametrelerden biridir. Buna rağmen proje yöneticileri hala efektif dış destekli yazılım maliyeti hesaplama metodları üzerinde yoğun bir şekilde çalışmaktadırlar. Bu alanda çok fazla metod geliştirilmiş olmasına rağmen bunlar arasında en uygunu seçmek hala önemli bir problemdir. Bir başka problem ise çok fazla yöntem olmasına rağmen bunların çoğunun sadece teoride kalması ve pratiğe dökülmemiş olmasıdır. Bütün bu sistemleri bir arada bulandıran SIP (software improvment proccess) frameworku geliştirilmiştir. Bu konuda çok fazla çalışma yapılmış olmasına rağmen *SIP frameworkunun* hala istenilen başarıyı yakalayamadığı gözlenmiştir Bunun ana kaynağının çift yönlü olduğu anlaşılmaktadır. Birinci proje yöneticilerinin deneyimleri ikincisi ise *SIP frameworkun'de* çok fazla metod olması ve bunlardan hala bazılarının pratikte çok fazla kullanılmamasıdır.

Yazılım geliştirmenin maliyetinin hesaplanması çok zordur. Geliştirme sürecinde sıkıntıya neden olan problemlerin çoğu bu sürecin hesaplanmasında karşılaşılan zorluklardan sorumludur. Herhangi bir hesaplamada ilk adımlardan birisi hesaplanacak sistemin anlaşılması ve tanımlanmasıdır. Bununla birlikte yazılım; somut olmayan, görünmez ve kontrol edilmesi zor bir yapıdadır. Görünmeyen ve dokunulmayan bir ürünün ya da sürecin anlaşılması ve hesaplanması doğal olarak daha zordur. Yazılım yazıldıkça gelişir ve değişir. Donanımsal bir tasarım yetersiz olduğunda veya bir donanım beklendiği gibi çalışmadığında, genelde yazılımda değişiklikler yaparak “çözüm” bulmaya çalışılır [3]. Bu değişiklikler geliştirme sürecinin sonlarına doğru olur ve bazen beklenmedik yazılım büyümesiyle sonuçlanır.

Son yirmi yılda yazılım maliyet tahmini için birçok model ortaya konmuştur. Bu modeller algoritma tabanlı modeller, analoji analizi modelleri, uzman kişi tahminleri yöntemi, aşağıdan yukarıya metodu ve yukarıdan aşağıya metodu gibi birçok metod geliştirilmiştir [4]. Hiç bir metod için birbirinden üstündür ya da kotudur gibi bir ayrıma gitmemek lazım. Her metodun zayıf ve güçlü yönleri mevcuttur. Geliştirilen projenin kategorisine göre özelleştirilmiş yöntemler mevcuttur. Bu nedenle modellerin zayıf ve eksik yönlerine bakarak metodların kombinasyonundan daha doğru sonuç veren bir model ortaya konabilir.

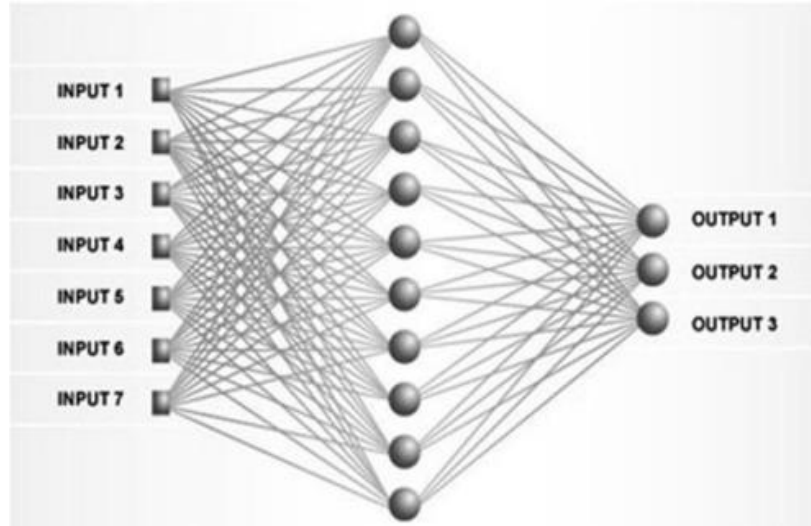
1.1. Yapay Sinir Ağları

Yapay Sinir Ağları (Artificial Neural Networks) veya kısaca Sinir Ağları (YSA) insan beyninden esinlenerek geliştirilmiş, ağırlıklı bağlantılar aracılığı ile birbirine bağlanan işlem elemanlarından oluşan paralel ve dağıtılmış bilgi işleme yapılarıdır. En önemli özelliği, deneyimlerden (tecrübe) yararlanarak öğrenebilmesidir. Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilmişlerdir. Yapay sinir ağları, öğrenmenin yanı sıra bilgiler arasında ilişkiler oluşturma yeteneğine de sahiptir [5].

Bir yapay sinir ağında, birbirleriyle bağlantılı sinir hücrelerinin yer aldığı girdi katmanı (input layer), çıktı katmanı(output layer) ve gizli katman(hidden layer) olmak üzere temelde üç katman bulunmaktadır. Girdi katmanı ilk katmandır ve dışarıdan gelen verilerin yapay sinir ağına alınmasını sağlar. Bu veriler istatistikte bağımsız değişkenlere karşılık gelmektedir. Girdi katmanı probleme etki eden parametrelerden oluşmaktadır ve girdi katmanındaki nöron sayısı parametre sayısına göre şekillenmektedir. Son katman çıktı katmanı olarak adlandırılır ve bilgilerin dışarıya iletilmesi işlevini görür. Çıktı değişkenleri, istatistikte bağımlı değişkenlere karşılık gelir [6]. Modeldeki diğer katmanlar ise girdi katmanı ile çıktı katmanı arasında yer alır ve gizli katman olarak adlandırılır. Gizli katmanda bulunan nöronların dış ortamla bağlantıları yoktur. Yalnızca girdi katmanından gelen sinyalleri alırlar ve çıktı katmanına sinyal gönderirler. Gizli katman ve gizli katmanlarda yer alacak nöronların sayı- sının seçimi, kurulan ağın performansı açısından önemlidir [7].

Yöntem, insan beyninin ‘nöron’ adı verilen birimlerin ağı şeklinde çalıştığı ile ilgili olan biyolojik bulgulardan esinlenmiştir. Tipik bir yapay sinir ağları modelinde girdi, gizli ve çıktı katmanları vardır. Şekil’1 de basit bir yapay sinir ağı modeli gösterilmiştir. Çıktı katmanı, girdi katmanından, aradaki gizli katmanlardan da geçerek, girdi ve sinyalleri sinyali alır. Gizli katman sayısı, uygulama alanına bağlı olarak değişiklik gösterir. Yapay sinir ağları, verimlilik tahmini, ses tanıma veya görüntü tanıma gibi verilen bir uygulamaya uygun olarak tasarlanırlar. Konu hakkında daha detaylı bilgi bu alanda yayınlanmış olan çok sayıda kitap ve çalışmadan elde edilebilir. [6 ve 7]

Şekil 1. Yapay Sinir Ağı Modeli



Temelde iki tür maliyet tahmin yöntemi vardır. Algoritmik ve algoritmik olmayan. Algoritmik modellerin bazıları istatistiği kullanarak basit aritmetik formüller kullanmaktadır. Algoritmik modeli bir takım değişkenleri kullanan bir fonksiyon olarak düşünebiliriz.

$$\text{Maliyet} = f(x_1, x_2, \dots, x_n) \quad (1)$$

x_1, x_2, \dots, x_n , maliyeti belirleyen (attribütler) faktörler olarak ifade edilebilir.

2. GELİŞTİRİLEM MODEL VE EFOR TAHMİNİ

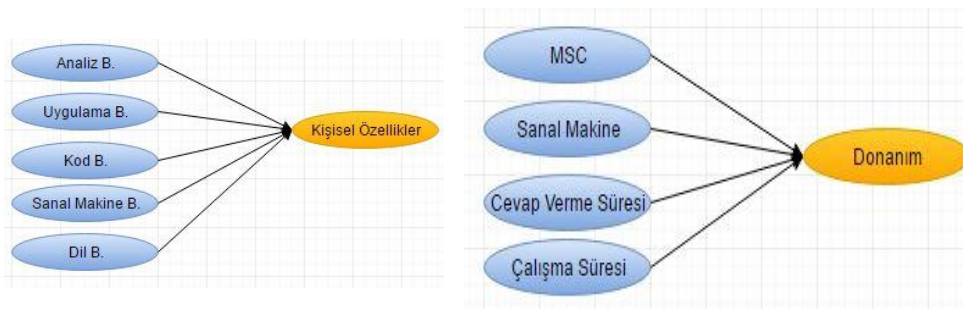
Birçok maliyet çalışmasında kullanılan COCOMO II modelinin 4 temel maliyet faktörü vardır: ürün faktörleri, bilgisayar faktörleri, personel faktörleri ve proje faktörleri. İkinci kısımda önerdiğimiz ölçek kümesi COCOMO'nun tüm ölçeklerini kapsamaktadır. Ancak bu 4 temel faktör yerine 6 temel faktör getirmiştir. Maliyet tahmini aslında oldukça karmaşık bir problemdir ancak bugüne kadar gereken ilgiyi görmemiştir. Araştırmacılar değişik yaklaşımlar getirmeye çalışmırlardır. Son zamanlarda yeni yeni yapay zekâ yaklaşımları düşünölmeye başlanılmıştır. Temel ve basit gösterimle durum Tablo 1'de gösterilmiştir. Yapay sinir ağları kullanılarak yapılan bazı çalışmalar vardır. Bu konu henüz yeni gelişen bir konudur. Bu konuda yapılmış çalışmalar Tablo 1'de listelenmiştir [8].

Tablo 1. YSA Tabanlı Çalışmalar

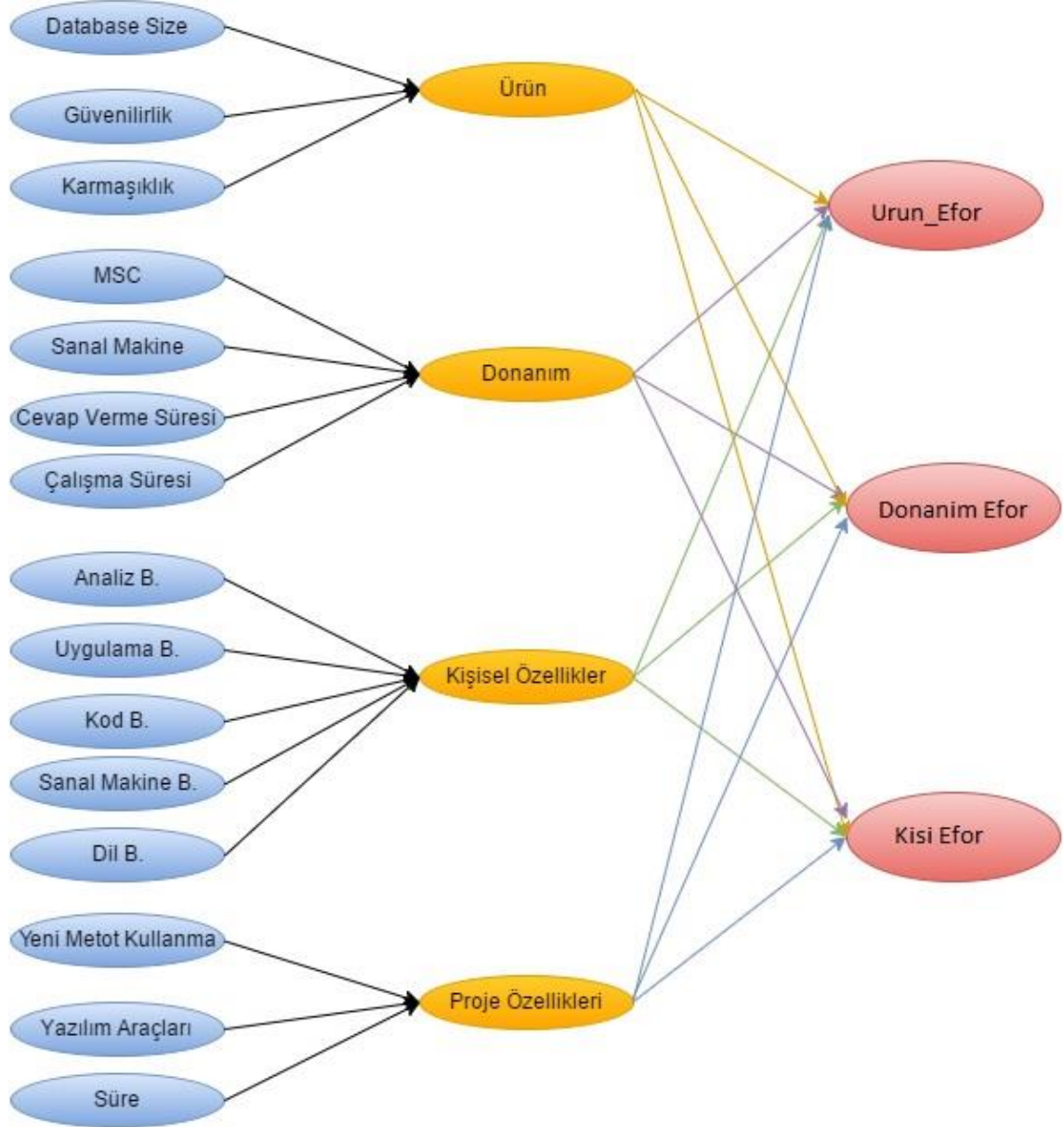
Çalışma	Kullanılan Algoritma	Ölçek Kümesi	Proje Sayısı	Sonuç (MMRE)
Wittig & Finnie [6]	Back-Propagation	ASMA ve Desharnais	136 ve 81	17%
Venkatachalm [7]	Back-Propagation	COCOMO	63	
Jorgenson [8]	Back-Propagation	Jorgensen	109	100%
Serluca [9]	Back-Propagation	Mermaid-2	28	76%
Karunanithi et al [10]	Cascade-Correlation			
Samson et al [11]	Back-Propagation	COCOMO	63	428%
Srinivasan & Fisher [12]	Back-Propagation	Kemerer ve COCOMO	78	70%
Hughes [13]	Back-Propagation	Hughes	33	55%

Şekil 2'de geliştirilen yapay sinir ağının bölüm bölüm hangi etmenlerden oluştuğu gösterilmiştir. Şekil 3 de yazılım maliyet tahminin tasarladığımız yapay sinir ağı modeli gösterilmiştir. Bu modele bağlı efor karşılık gelen girdi değerleri ilgili veri setlerinden alınarak gösterilmiştir. Önemli olan ara katmanların oluşturulması ve sonuncularının belirlenmesidir.

Şekil 2. Geliştirilen Basit Yapay Sinir Ağı Modeli (bolum bolum)



Şekil 3. Geliştirilen Yapay Sinir Ağı Modeli



Buradaki projelerin değerleri bizim belirlediğimiz niteliklere göre gerçek değer çıktı daha sonra bu verileri modelimize göre uyguladık. Tablo 3’de ifade edilen değerleri elde ettik. Elde ettiğimiz değerler gerçek değerlere oldukça yakın olduğunu görebilmekteyiz.

2.1. Veri Seti

PROMISE sitesi üzerinde, açık olarak ulaşılabilecek efor tahmininde kullanılacak veri kümeleri mevcuttur. Önerilen model için bu kümelere COCOMONASA veri kümesi kullanılmıştır. Küme içerisinde 94 satır bulunmaktadır. Çalışmada daha sağlıklı sonuç elde edebilmek için tüm değerler normalize edilmiştir. Normalizasyondan sonra sınır değerlere sahip olan minimum ve maksimum satırları çıkarılmıştır.

Küme, çalışmalarda 2 bölüme ayrılmış, %75' lik bölümü eğitim için, kalan ise test için kullanılmıştır. Kümelerin oluşturulması için bir random fonksiyon kullanılmıştır. Bu fonksiyona göre, tüm küme içerisinden belli sayıdaki satırlar 2 yeni alt kümeyi oluşturmuştur. Özellikle eğitim aşamasında veri kümesinin içeriği ile beraber, verilerin sırası da sonuçları etkileyebileceğinden, daha modüler bir yapıya ulaşmak için rastgele elemanların seçilmesi önemlidir. Bilhassa k-means algoritmasında başlangıç değerleri ve kümeler, sonuçta oluşan alt kümeleri çok fazlaca etkilemektedir. Bu nedenle eğitim ve test veri kümelerinin oluşturulmasında sıra gözetilmeden, rastgelelik sağlayan metot tercih edilmiştir [9].

Veri kümesinde kullanılan özelliklerin kısaltmaları ve açıklamaları aşağıdaki gibidir.

RELY: (Required Software Reliability)Yazılım güvenilirliğini, istenilen fonksiyonlarını en yakın çalışma zamanında tatmin edici bir şekilde yerine getirmesi olasılığı.

DATA: (Database Size)Geliştirilen yazılım ürününün veritabanı büyüklüğüdür. Bir yazılım ürününü geliştirmek için gereken işgücü miktarı, açıkça veritabanı ile doğrudan alakalıdır.

CPLX: (Software Product Complexity)Geliştirilecek modül karmaşıklığı.

TIME: (Execution Time Constraint)Bir yazılım alt sistemlerindeki çalışma süresi.

STOR: (Main Storage Constraint)Bir yazılım alt sistemlerindeki depolama alanı.

VIRT: (Virtual Machine Volatility)Geliştirilecek yazılım ürününü altında yatan sanal makinenin oynaklık düzeyini yansıtır. Sanal makine ürününün görevlerini yerine getirmesi için üzerinde çalıştıracağı donanım ve yazılım kompleksi olarak tanımlanır.

TURN: (Computer Turn Around Time)Geliştirilen sistem tarafından kullanılan bilgisayarların cevap verme süresi.

ACAP: (Analist Capability)Yazılım alt katmanlarında çalışanların analist kapasitesi.

AEXP: (Application Experience)Yazılımı gerçekleştiren takımın uygulama becerisi.

PCAP: (Programmer Capability)Programcının kod becerisi.

VEXP: (Virtual Machine Experience)Yazılımı gerçekleştiren takımın kullanılacak sanal makine deneyimi.

LEXP: (Language Experience)Yazılımı gerçekleştirecek takımın kullanılacak programlama dili tecrübesi.

MODP: (Use Of Modern Programming Practices)Yazılım kullanılırken yeni çıkan gelişmelere ait uygulamaları kullanma becerisi.

TOOL: (Use Of Software Tools)Yazılım da kullanılan araçlar.

SCED: (Development Schedule Constraint)Yazılım aşamalarının her birinin gerçekleştirmesi için verilen kısıtlanmış süre

Tablo 2. YSA tabanlı Efor Tahmini

Pr.No.	Boyut	Gerçek Değer	Yapay Sinir Ağları
1	002.00	002.00	002.98
2	114.28	018.00	281.00
3	064.10	332.00	246.80
4	023.11	004.00	065.40
5	001.37	001.00	000.94
6	001.61	002.10	002.00
7	031.85	005.00	145.20
8	131.00	619.90	725.30
9	010.00	003.00	037.40
10	015.00	004.00	064.30
11	004.25	004.50	008.40
12	004.05	002.00	003.20
13	019.90	074.60	091.60
14	003.00	001.20	002.80
15	040.53	022.00	024.20

3. SONUÇLAR VE DEĞERLENDİRMELER

Efor tahmini proje yönetiminde en önemli konulardan biridir. Bu süreci destekleyecek araç ve yöntemlerin ortaya konması, proje yöneticilerinin daha başarılı tahmin yapmalarını ve dolayısıyla daha iyi proje yönetimine sahip olmalarını sağlayacaktır. Bu çalışmada, efor tahmininde farklı bir yaklaşım olarak, üst ve alt limitlere göre efor tahmini yapılmış ve önerilen model, test verileri ile denenmiştir. Sonuçlardan yapılabilecek çıkarımlara göre önerilen modelin dengeli olduğu, amaca hizmet edebileceği, karar destek aşamalarında birden çok çıktı ile yorum yapmanın daha sağlıklı olabileceği gösterilmiştir. Model, COCOMO üzerine bina edilmiş ve YSA algoritması ile kullanılmıştır. Model sonuçlarının tek başına hem COCOMO hem de YSA' dan daha başarılı olduğu gözlemlenmiştir. Çalışma, COCOMONASA adlı veri kümesi ile yapılmıştır ve bu kümede 94 satır bulunmaktadır. Modelin veri kümesinin büyüklüğünden etkilenip etkilenmediği bilinmemektedir. Aynı zamanda, limit değerleri basitçe bir sıralama fonksiyonu sonrası, YSA tahmin değerine en yakın satırların alınması şeklinde hesaplanmıştır. Proje karakteristiğine göre bu hesaplamanın daha esnek bir yapıya kavuşturulması, sonuç değerlerini daha anlamlı hale getirebilir. Ek olarak çalışmada standart YSA kullanılmış ve herhangi bir bias nöron dâhil edilmemiştir. Bias değerinin sisteme eklenmesi ve limit değerlerinin aralığının daha modüler şekilde hesaplanması olası gelecek çalışmaları arasında sayılabilir.

Bu çalışmada efor kestirim yöntemleri arasında en çok kullanılan COCOMO modelinden faydalanılarak, günümüzde yaygın olarak kullanılan yazılım geliştirme için pratik bir efor kestirim modeli oluşturulmuştur. COCOMONASA veri setiyle hazırlanan küçük ve orta ölçekli proje verileri ile gerçekleştirilen bu çalışma sonucunda, modelin ortalama %6 oranında sapma ile gerçeğe yakın tahmin yapabildiği görülmüştür. Bu oran yapılan benzer çalışmalara göre oldukça düşük bir orandır. Örneğin Z. Zia, A. Rashid ve K. uz Zaman'ın 2009 yılında geliştirdiği modelle Vb.Net, Visual Basic ve Visual C# ile geliştirilen 19 proje için %11.61 MMRE değeri elde edilebilmiştir. Van Koten modeli bu projeler için uygulandığında ise

%17.81 MMRE değeri elde edilebilmiştir. Modelin çekirdek veri kümesini oluşturan 15 projeden dış kaynak kullanılarak gerçekleşen 13 tanesinin farklı firmaların farklı ekipleri tarafından gerçekleştirilmiş olması, modelin geçerliliğine olumlu katkıda bulunmuştur. Bu modelin farklı ortamlarda ve farklı organizasyonlarda başarılı sonuçlar sunabilmesi için her ortam için ayrı kalibrasyon yapılmasına ihtiyaç olduğu görülmüştür. Elde edilen sonuçların başarısının seçilen nesne sayılarının ve bunlara ait karmaşıklık derecelerinin doğru bir şekilde seçilmesi ile doğru orantılı olarak arttığı görülmüştür

KAYNAKLAR

- [1] B. W. Boehm, Software Engineering Economics, Prentice Hall, 1981.
- [2] A. J. Albrecht and J. E. Gaffney, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," Proc. IEEE Transactions on Software Engineering, IEEE Press, Nov. 1983, pp. 639-648.
- [3] R.D.H. Warburton, "Managing and Predicting the Costs of Real-Time Software," IEEE Trans. Software Eng., vol. 9, no. 5, pp. 562-569, Sept. 1983.
- [4] Krishnakumar Pillai, V.S. Sukumaran Nair, "A Model for Software Development Effort and Cost Estimation", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 23, NO. 8, AUGUST 1997
- [5] N. Tadayon, "Neural Network Approach for Software Cost Estimation," Proc. IEEE The International Conference on Information Technology: Coding and Computing (ITCC'05), IEEE Press, Apr. 2005, pp. 815-818.
- [6] B. T. Rao, "A Novel Neural Network Approach For Software Cost Estimation Using Functional Link Artificial Neural Network (FLANN)," International Journal of Computer Science and Network Security, vol. 9, Jun. 2009, pp. 126-131.
- [7] Kultur, Y., Turhan, B. ve Bener, A.: Ensemble of neural networks with associative memory (ENNA) for estimating software development costs, Knowledge-Based Systems, Cilt 22, No 6, pp. 395–402 (2009)
- [8] Riquelme, J.C., Polo, M., Aguilar Ruiz, J.S., Piattini M., Ferrer Troyano, F. J., Ruiz, F., "A Comparison of Effort Estimation Methods for 4GL Programs: Experiences with Statistics and Data Mining". Int'l. J. of Software Engineering and Knowledge Engineering, vol. 16(1), 2006
- [9] Marcus, A. ve Menzies, T.: Software Is Data Too , International Symposium on Foundations of Software Engineering, Santa Fe, NM, ADB., 229-232 (2010)