



Bilinmeyen PLC Programının Makine Öğrenmesi Yöntemleriyle Taklit Edilmesi

Imitating of an Unknown PLC Program with Machine Learning Methods

Faruk Ulaşım^{*1}, Yasin Yüce², Bülent Cesur³

¹Kırıkkale Üniversitesi Hacılar Hüseyin Aytemiz MYO, 71010 Kırıkkale, TÜRKİYE

²Sanayi ve Teknoloji Bakanlığı, 58020 Sivas, TÜRKİYE

³Çalışma ve Sosyal Güvenlik Bakanlığı, 06520 Ankara, TÜRKİYE

Başvuru/Received: 22/09/2023

Kabul / Accepted: 31/10/2023

Çevrimiçi Basım / Published Online: 31/12/2023

Son Versiyon/Final Version: 31/12/2023

Öz

Programlanabilir Mantık Denetleyiciler (PLC) uzun yıllardır endüstrinin hemen her alanında kullanılmaktadır. Kullanılan bu PLC'lerin eskimesi, bozulması veya şifreyle korunması gibi durumlarda PLC programlarının yedeklenmesi mümkün olmamaktadır. Herhangi bir arıza sonucu PLC programının silinmesi durumunda veya PLC'nin yenilenmesi ihtiyacı olduğunda programın yeniden yazılması gerekir. Böyle bir durumda sistemin çalışma adımlarının detaylı bir şekilde bilinmesi gerekir ve program yazılması esnasında atlanacak bir adım sistemin tamamında çok büyük problemlere sebep olabilir. Bu çalışmada, çeşitli makine öğrenmesi algoritmaları kullanılarak PLC içerisinde çalışan ve bilinmeyen bir programın çalışma adımlarının taklit edilmesi işlemi yapılmıştır. Bunun için ilk olarak bir veri günlüğü oluşturularak PLC'nin giriş ve çıkış bilgileri kaydedilmiştir. Daha sonra bu giriş-çıkış verileri Makine Öğrenmesi algoritmaları ile eğitilmiştir. Eğitilen bu algoritmaların giriş veri setine karşılık verdiği çıktıları, PLC çıktılarıyla paralel olarak izlenmiştir. Makine öğrenmesi algoritması olarak karar ağacı, k-en yakın komşu ve rastgele orman algoritmaları kullanılmıştır. Algoritmaların performans ölçüm metriği olarak doğruluk puanı (accuracy score) kullanılmıştır. Yapılan çalışmalar sonunda Rastgele Orman algoritmasının daha iyi sonuç verdiği gözlemlenmiştir.

Anahtar Kelimeler

“Programlanabilir Mantık Denetleyiciler, Makine Öğrenmesi, Bilinmeyen PLC Programının Modellenmesi”

Abstract

Programmable Logic Controllers (PLC) have been used in almost every field of industry for many years. It is not possible to back up PLC programs in cases such as aging, deterioration or password protection of these PLCs. If the PLC program is deleted as a result of any malfunction or when the PLC needs to be replaced, the program must be rewritten. In such a case, the working steps of the system must be known in detail, and a step that is skipped during the writing of the program can cause major problems in the entire system. In this study, various machine learning algorithms are used to imitate the execution steps of an unknown program running in a PLC. For this purpose, a data log was first created and the input and output information of the PLC was recorded. Then, these input-output data were trained with Machine Learning algorithms. The outputs of these trained algorithms in response to the input data set were monitored in parallel with the PLC outputs. Decision tree, k-nearest neighbor and random forest algorithms were used as machine learning algorithms. Accuracy score was used as the performance measurement metric of the algorithms. At the end of the studies, it was observed that the Random Forest algorithm gave better results than the Machine Learning algorithms.

Key Words

“Programmable Logic Controller, Machine Learning, Modeling of an Unknown PLC Program”

1. Giriş

Sanayi devriminin başlaması ile birlikte insan gücünün yerini alan ve üretimi hızlandıran çok çeşitli makineler geliştirilmiştir. Üretilen bu makineler için ortaya çıkan ilk problem otomatik çalışma ihtiyacı olmuş ve bu ihtiyaç röleli kontrol devreleriyle (Relay Ladder Logic -RLL) çözülmüştür (Erikson, 1996). Rölelerin açık ve kapalı kontaklarının kullanılması ile kurulan kontrol devreleri çok uzun yıllar endüstrinin çeşitli alanlarında kullanılmıştır (W. Bolton, 2015). Otomatik kontrol ile çalışan cihazlarda büyük bir kolaylık sağlayan RLL' li sistemler, özellikle birkaç farklı çalışma moduna sahip makinelerde, her çalışma modu için RLL devrelerinin tamamının değiştirilmesi gerekliliği bu sistemi çok dezavantajlı bir hale getirmiştir. Bu durum 1960 lı yılların sonunda programlanabilir mantık denetleyicilerin (PLC) üretilmesiyle son bulmuştur (L de MBA Dib, 2017). PLC ile kurulan devreler RLL ile kurulan kontrol devreleriyle karşılaştırıldığında önemli avantajlarının olduğu görülmüştür. Bu avantajlardan bazıları;

- Kablo kullanımı %50 den fazla olacak şekilde azalmıştır,
- Zamanlayıcı, sayıcı gibi ekstra ekipmanların kullanılmasına gerek kalmamıştır,
- Güç tüketimi önemli ölçüde azalmıştır,
- Kontrol akışını değiştirilmesi gerektiği durumlarda tüm sistemin değişmesine gerek kalmamıştır,
- Kötü çevresel koşulların çalışan sistem üzerindeki etkisi azalmıştır,
- Operasyon hızı milisaniyeler seviyesine düşmüştür. (Zimmerman, 2008)

PLC, röleli kontrol sistemlerindeki “merdiven diyagramı” mantığını taklid eden, bilgisayar tabanlı ve tek işlemcili elektronik ekipmanlardır (A. Rullan, 1997). İlk olarak Dick Morley ve arkadaşları tarafından 1968 yılında üretilmiştir (B.C. Brusso, 2018). Ortaya çıktığı ilk günden beri yoğun bir şekilde kullanılan PLC' ler günümüzde de endüstrideki otomatik kontrol sistemlerinin temel ekipmanıdır (R. Bayındır, & Y. Cetinceviz, 2011). Endüstriyel otomasyonun temelini oluşturan PLC' ler kullanıldıkları sistemlerde üç temel işlevi yerine getirir. Bunlar; otomasyonu yapılan sistemin süreçlerini izlemek ve kontrol etmek, gerekli yerlerden ölçüm alma ve test etme ile ilgili görevleri yerine getirmek ve son olarak sürekli çalışan sistemlerde süreçleri birbirini takip edecek şekilde yürütmektir (M. Dobosz, 2020).

PLC'ler, bahsi geçen bu üç temel işlevi yerine getirmek için çalışma akışına ve mantığına dayalı olarak karar vermesi için ihtiyaç duyduğu verileri sensörler ve diğer giriş ekipmanları aracılığıyla alır. Bu giriş ekipmanlarından aldıkları sinyalleri, içerisinde programa uygun olacak şekilde periyodik taramalar ile kontrol ederek çıkış elemanlarını aktif eder ve bu işlemleri sürekli döngüler şeklinde tekrar ederek girişlerin durumuna göre çıkışların durumunu günceller (Özerdem, 2016) Bu kullanım şekliyle düşünüldüğünde hem belleğini hemde giriş-çıkış verilerini işleyen bir “gömülü sistem” olarak tanımlanabilir (J.I. Lee, 2002).

PLC programlama için çeşitli programlama dilleri mevcuttur. Bu programlama dillerinin standart hale getirilmesi için ilk olarak Aralık 1993'te IEC 61131 açık uluslararası standart geliştirilmiştir. Mevcutta Şubat 2013'te yayınlanan 3. baskı bulunmaktadır (B.G. Silva & M. De Sousa, 2016). Bu programlama dilleri seti, endüstri tarafından geniş çapta benimsenmiştir ve endüstriyel otomasyon sistemlerinin çoğu bu dilleri temel almaktadır (D. Soliman, 2012). IEC standardı üç grafiksel ve iki metinsel programlama dili standardı tanımlar. Bunlar; Merdiven Diyagramı (Ladder Diagram) (LD), Fonksiyon Blok Diyagramı (Function block diagram) (FBD), Komut Listesi (Instruction list) (IL), Yapılandırılmış Metin (Structured text) (ST), Sıralı Fonksiyon Tablosudur (Sequential function chart) (SFC).

PLC içerisine yazılan programlar çeşitli şekillerde yazılabilesine rağmen en çok kullanılan yöntem “Merdiven Diyagramı” (LD) yöntemidir (Liu, J., & Darabi, H., 2002). LD, röleli kontrol sistemleriyle yapılan devrelere çok benzemektedir. Bu benzerlikten dolayı, otomatik kontrolün temelini oluşturan röleli kontrolden LD' ye geçiş çok kolay olmaktadır. Bunun yanında PLC' nin PC tabanlı bir ekipman olmasından dolayı, PLC içerisine çeşitli komutlar entegre edilerek program yazılmasının kolaylaştırılması sağlanmıştır. PC tabanlı olması ve program yazmayı kolaylaştıracak komutlar kullanılmasına rağmen uzun bir programın anlaşılması ve hata ayıklaması gibi işlemlerin yapılması zor olmaktadır (Guasch, A., 2000). Çünkü genellikle LD ile yazılan programlarda, program kodlarının tamamı tek bir ekran üzerinden görüntülenemez. Sistemin aksadığı noktanın tespit edilmesi için tüm program satırları adım adım kontrol edilmek durumunda kalınabilir. Bu durum ise arıza durumlarının uzamasına sebep olur. Bu ise işletme maliyetlerini artıran bir durumdur.

Hata ayıklama işleminin daha kolay olması için Durum Diyagramları (Otto, H. P., & Rath, G. (1996), Akış Şemaları (Lobov, A., 2003), Belirleyici Otomatlar (Carayol, A., & Nicaud, C., 2012), Petri Ağları (Kaid, H., 2020) vb gibi yöntemler kullanılır. Bu yöntemler sayesinde, LD ile programı yazılmış olan PLC programlarının görsel bir şekilde görülmesi sağlanır ve bu durum hata-arıza arama gibi işlemleri oldukça kolaylaştırır. Ayrıca bu yöntemler ayrık olayları içerdiği için, programın akış işlemi görselleştirildikten sonra LD' ye geçiş oldukça kolay olmaktadır. Her ne kadar LD ile yazılmış bilinen bir PLC programından bahsi geçen yöntemler kullanılarak programın akışının görselleştirilmesi sağlanabilse de bilinmeyen bir PLC programının giriş ve çıkışlarının izlenerek akış şemasının yazılması, program içerisinde özellikle sayıcı ve zamanlayıcı varsa oldukça zorlayıcı bir durumdur ve literatürde bu konularla ilgili çalışmalar devam etmektedir.

Kullanım kolaylıkları, güvenilir çalışma şekilleri ve her tür endüstriyel ortama uyum sağlayabilme özelliklerinden dolayı uzun yıllardır kullanılan PLC' ler; günümüzdeki internet tabanlı sensörlerle ve "büyük veri" sistemlerine olan uyumluluğundan dolayı Endüstri 4.0 gibi sistemlerinde vazgeçilmez bir ekipmanı olmaya devam edecektir (Wan, 2016). Bunun yanında endüstriye 1960' lı yıllarda girmesi ve yıllar içerisinde kullanılan yazılımların, dokümantasyonların ve işletim sistemlerinin değişmesi gibi sebeplerden dolayı belirli zaman aralıklarında güncellenme ihtiyacı duyulmaktadır (Barthelmey, 2014). Yazılımların eskimesi, işletim sistemlerinin yenilenmesi gibi sebepler PLC programına ulaşmayı engellemektedir. Bunun yanında bazı PLC' lerin programlarına şifre konulması veya işlemcinin zarar görmesi gibi durumlarda da ulaşılamamaktadır. Bu durumlarda sistemin programının yeni baştan yazılması gerekmektedir.

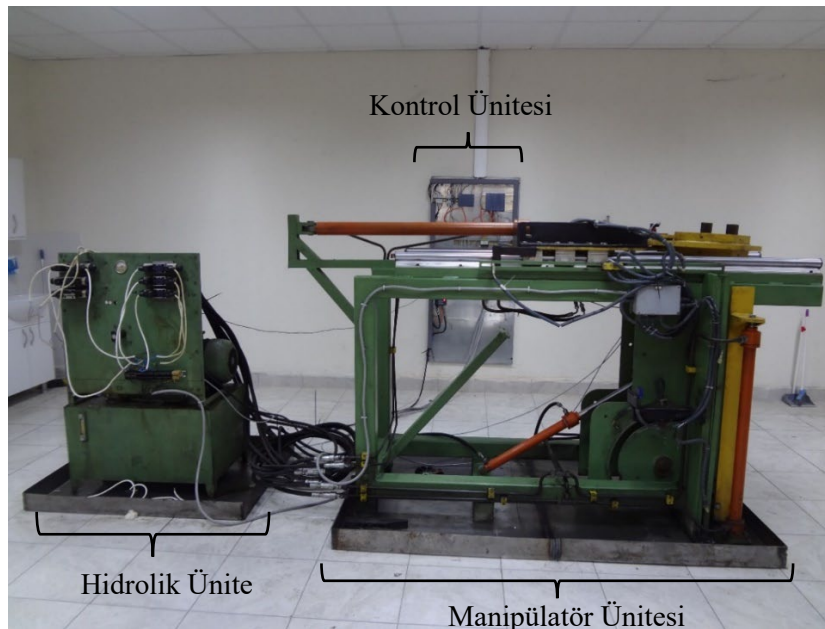
Bu çalışmada, PLC içerisinde çalışan ve akış şekli bilinmeyen bir programın çalışma adımlarının taklit edilmesi için makine öğrenmesi modelleri kurulmuştur. Bu bağlamda ilk olarak bir veri günlüğü oluşturularak sistemin normal çalışması esnasındaki PLC' nin giriş ve çıkış verileri kaydedilmiştir. Elde edilen giriş verilerine karşılık gelen çıkış verileri çeşitli makine öğrenmesi algoritmasından geçirilmiş ve giriş durumlarına karşılık gelen çıkış değerlerinin öğrenilmesi sağlanmıştır. PLC programında analog sensör ve aktüatörler kullanılmamış; sadece dijital giriş-çıkış elemanları kullanılmıştır. PLC çıkışı 0 veya 1 olacağı için makine öğrenmesinde "sınıflandırma yöntemleri" üzerinde durulmuştur. Ayrıca kullanılan deney düzeneği üzerinde 8 giriş ve 8 çıkış elemanı olduğu için çoklu sınıf-çoklu çıkış sınıflandırmaya uygun olan karar ağacı, k-en yakın komşu ve rastgele orman algoritmaları uygulanmıştır. Yapılan çalışmanın uygulanması için, hidrolik bir presin manipülatör ünitesi test düzeneği olarak kullanılmıştır. Test düzeneği üzerinde çeşitli zaman dilimlerinde veriler alınarak makine öğrenmesi algoritması verileri elde edilmiştir. Test düzeneği ve sistemin çalışması için kullanılan makine öğrenmesi algoritması ile ilgili detaylar github¹ da internete açık olarak yayınlanmıştır.

2. Materyal

Endüstriyel ortamlarda yüksek sıcaklık veya ağırlıktaki malzemelerin taşıma işlemleri genellikle endüstriyel robotlar veya manipülatörler vasıtasıyla yapılır. Robotlar, gövde boyutlarına oranla uzak bir mesafeye taşıma işlemi yaparlarken; manipülatörler taşıma işlemini birden fazla eklem birbiriyle bağlantılı kollar aracılığıyla yapar (Sönmez, 2005). Manipülatörlerde sistemin yük gereksinimleri göz önüne alınarak; servo motorlar, hidrolik sistemler veya pnömomatik sistemler kullanılabilir. Bu sistemlerin birbirleri üzerinde çeşitli avantajları olsa da özellikle yüksek ağırlıklı ekipmanların taşınması genellikle hidrolik sistemler vasıtasıyla gerçekleşir. Bu çalışmada hidrolik sistemle çalışan bir manipülatör, test ekipmanı olarak kullanılmıştır. Kullanılan test ekipmanı ile ilgili detaylar ilerleyen kısımlarda verilmiştir.

2.1. Tasarlanan Test Düzeneği

Tasarlanan test düzeneği Şekil 1' de görüldüğü gibidir. Bu deneysel sistem; sıcak presleme işlemi yapılan hidrolik presler arasında malzeme taşıma görevini yapan hidrolik bir ekipmandan çıkartılarak yenilenmiş ve deneysel çalışmalarda kullanılmak üzere yeniden düzenlenmiştir. Bu sistemde bulunan giriş ve çıkışlar Tablo 1' de görüldüğü gibidir.



Şekil 1. Tasarlanan Test Düzeneği

¹ https://github.com/mesleki2017/plc-laboratuvar/blob/master/colab/sep_02.ipynb

Tablo 1. Kullanılan PLC' nin Giriş-Çıkış Adresleri

Sıra No	Kayıt İsmi	Açıklama	PLC Veri tipi	PLC Adresi
1	Araba İleride	Taşıyıcı Araba Geride	Dijital Giriş	I0.0
2	Araba Geride	Taşıyıcı Araba İleride	Dijital Giriş	I0.1
3	Çene-açık	Tutma Çenesi Açık	Dijital Giriş	I0.2
4	Çene-kapalı	Tutma Çenesi Kapalı	Dijital Giriş	I0.3
5	Devirme-yukarıda	Devirme Tertibatı Yukarıda	Dijital Giriş	I0.4
6	Devirme-asağıda	Devirme Tertibatı Aşağıda	Dijital Giriş	I0.5
7	Kaldırıcı-asağıda	Kaldırma Tertibatı Aşağıda	Dijital Giriş	I0.6
8	Kaldırıcı-yukarıda	Kaldırma Tertibatı Yukarıda	Dijital Giriş	I0.7
9	Kaldırıcı-yukarı	Kaldırma Tertibatı Yukarı	Dijital Çıkış	Q0.0
10	Kaldırıcı-asağı	Kaldırma Tertibatı Aşağı	Dijital Çıkış	Q0.1
11	Araba-ileri	Taşıyıcı Araba İleri	Dijital Çıkış	Q0.2
12	Araba-geri	Taşıyıcı Araba Geri	Dijital Çıkış	Q0.3
13	Çene-kapa	Tutma Çenesi Kapama	Dijital Çıkış	Q0.4
14	Çene-aç	Tutma Çenesi Açma	Dijital Çıkış	Q0.5
15	Devirme-yukarı	Devirme Tertibatı Yukarı	Dijital Çıkış	Q0.6
16	Devirme-asağı	Devirme Tertibatı Aşağı	Dijital Çıkış	Q0.7

Otomatik kontrol ile yapılan işlemler birbirini takip eden operasyonlar şeklinde devam eder. Bir işlemi sona erdiren bir olay, bir diğer işlemin başlangıcını oluşturur. Böylece birbirinden bağımsız kesikli olaylar dizisi, birbiri ardınca devam eden sıralı olaylar şeklinde sistemin dinamik davranışını belirler (Glavaš, 2021). Bu çalışmada kullanılan test düzeneği, hidrolik ünite üzerinde bulunan yön valfleriyle kesikli olaylar dizisi şeklinde kontrol edilmektedir. Yön valfleri PLC tarafından gönderilen çıkış sinyalleri vasıtasıyla pistonlara hareket vermektedir. Ayrıca her bir hareketin sonunda o işleme ait bir giriş sensörü bulunmaktadır. Bu giriş sensörleri sayesinde yapılması gereken hareketin tamamlandığı ve diğer harekete geçilebileceği bilgisine ulaşılmaktadır. Bu bağlamda test ekipmanı hem tek bir döngü ile çalışmaya hemde otomatik olarak sürekli çalışmaya uygun bağımsız kesikli olaylar dizisine uygundur. Sistem tahriki güçlü bir hidrolik ünite aracılığıyla verildiği için, sistemin algoritmasının doğru sırayla çalışması çok önemlidir. Çalışma sırasındaki herhangi bir sıra değişikliği tüm sistem için onarılması zor bir mekanik arızasına sebep olabilir. Tasarlanan test düzeneği için çalışma durum diyagramı Şekil 2'de görüldüğü gibidir.

Tasarlanan test düzeneği, Şekil 1'de görüldüğü gibi üç kısım üzerinde incelenebilir. Bu ünitelerden manipülasyon ünitesi sistemin temel işlevine uygun bir şekilde mekanik olarak tasarlanmış, tüm sistemin kontrolü ise PLC tabanlı bir kontrol ünitesi tarafından kontrol edilmektedir.

2.1.1. Manipülasyon ünitesi

Bu ünite üzerinde 5 adet hidrolik piston, bu pistonların sınırlarının kontrol edildiği 8 adet endüktif sensör, bir adet ileri hareketin kontrol edildiği incremental lineer encoder (lineer cetvel), bir adet aşağı yukarı hareketin kontrol edildiği analog sınır sensörü (analog cetvel) bulunmaktadır. Bu sensörlerden incremental lineer cetvel ve analog sınır sensörü bu çalışma kapsamında olmadığı için program içerisinde kullanılmamıştır.

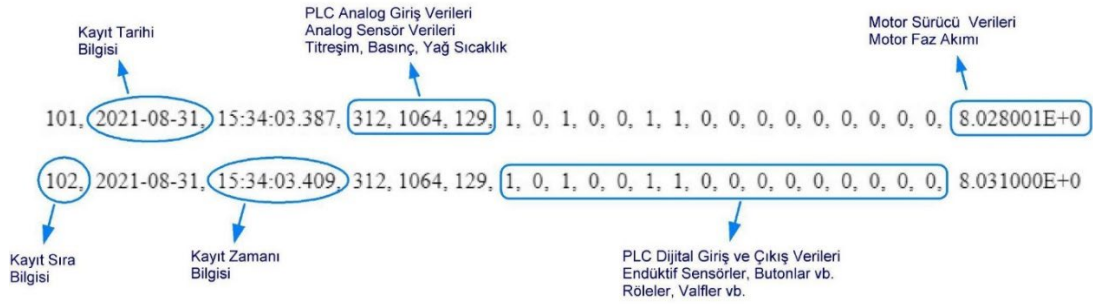
2.1.2. Hidrolik ünite

Hidrolik ünite, manipülasyon pistonlarının hareketleri için gerekli olan hidrolik yağın ve yön valflerinin bulunduğu bölümdür. Kontrol ünitesinden gelen komutlara göre valfler vasıtasıyla manipülasyon ünitesine hareket veren pistonların kontrol edilir. Hidrolik ünite için gerekli güç, test düzeneğine eklenmiş bir adet 5.5 KW asenkron motor ile verilmektedir. Bu motorun yanında ayrıca bir adet hidrolik pompa ve 6 adet yön valfi bulunmaktadır.

2.1.3. Kontrol ünitesi

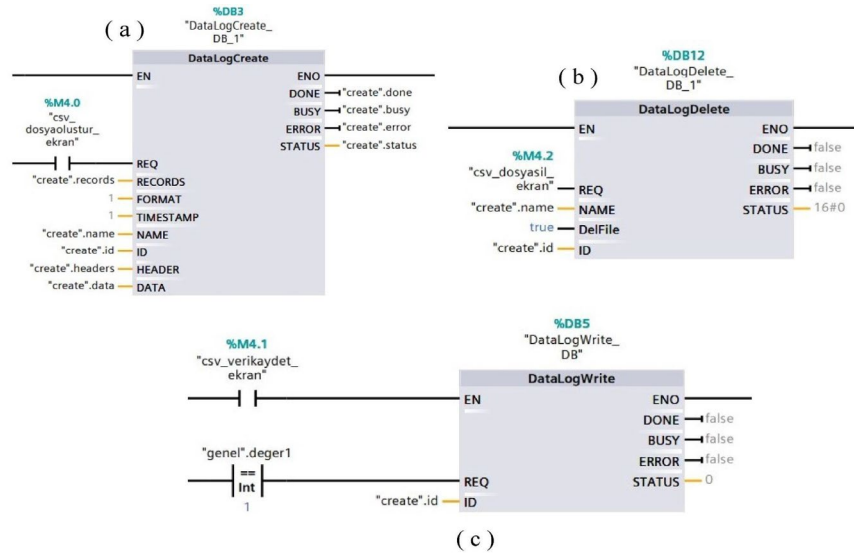
Tüm deney setinin kontrolünün yapıldığı kısımdır. Kontrol işlemi bir PLC vasıtasıyla yapılmıştır. PLC olarak S7-1215 CPU kullanılmıştır. PLC nin giriş ve çıkışlarının bilgisinin veri günlüğünde oluşturulması için bir adet hafıza kartı bulunmaktadır. Çalışma esnasında hafıza kartına kaydedilen giriş-çıkış verileri .csv uzantılı olarak bilgisayara alınmıştır.

Bu donanım ve yazılımsal alt yapı sayesinde kurulan ağ üzerinden PLC giriş-çıkış veri kayıtlarına “.csv” dosyası olarak erişim sağlanmıştır. Dokümanın uzantısının ne olacağı ve hangi verileri kaydetmesi gerektiği ile ilgili ayarlar Şekil 5’ de görüldüğü gibidir. Bu çalışmada makine öğrenmesi uygulaması Python programı ile eğitileceği için csv dosya şekli belirlenmiştir. Ayrıca, Şekil 4’de kurulan ağ üzerinden çekilen bir csv dosyası ve bu dosyadaki verilerin türleri hakkında bilgi görülmektedir.



Şekil 4: Veri kayıt açıklamaları

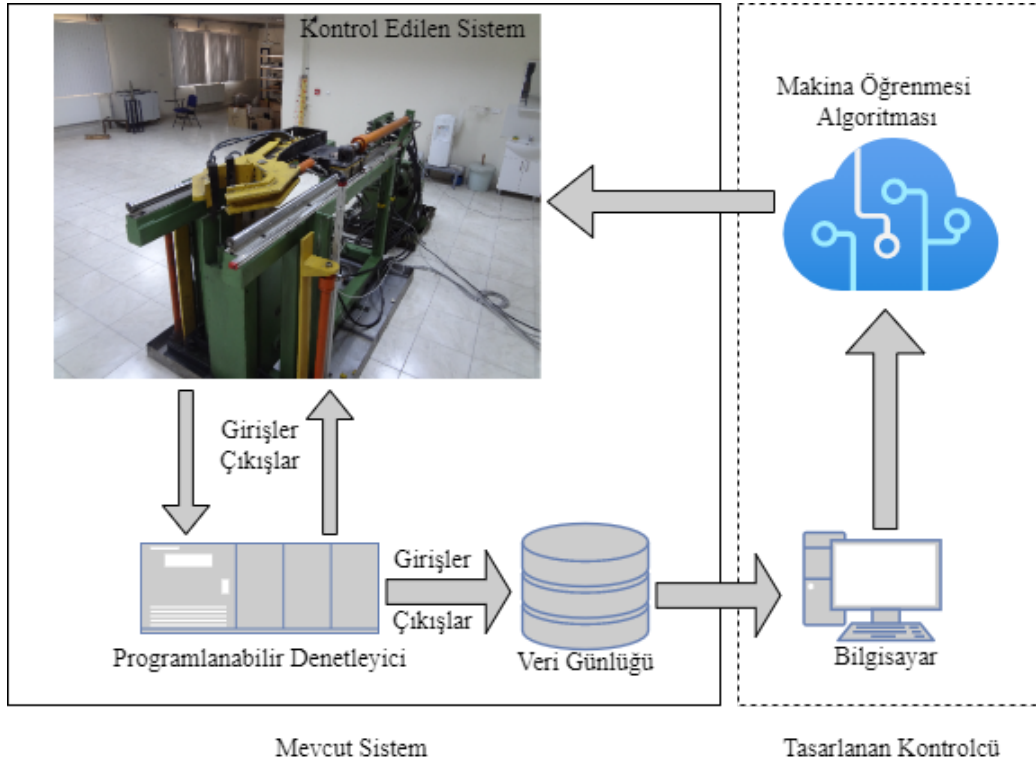
PLC giriş-çıkış kayıtlarının oluşturulabilmesi için Siemens TIA Portal üzerinden çeşitli fonksiyonlar kullanılmıştır. Bu fonksiyonlar Şekil 5’te görüldüğü gibidir.



Şekil 5: (a) PLC programında .csv veri kayıt dosyası oluşturma fonksiyonu, (b) PLC programında veri kayıt dosyası silme fonksiyonu, (c) PLC programında veri kayıt dosyasına istenilen veriler yazma fonksiyonu

3. Metot

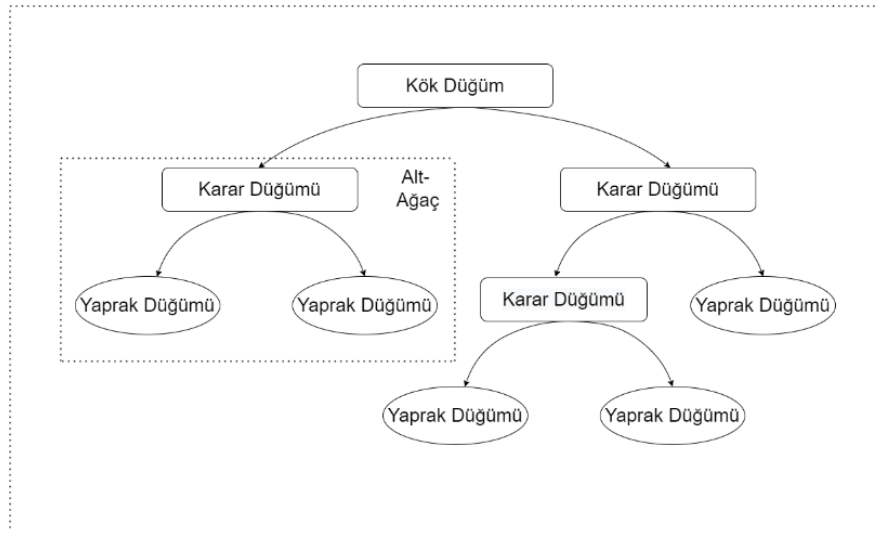
Bu çalışmada, makine öğrenmesi yöntemi ile PLC programının taklit edilmesi işlemi için kullanılan deneysel sistem tasarlanmıştır. İlk olarak sistemin giriş ve çıkış bilgileri bir veri günlüğü ile kaydedilmiştir. Kaydedilen bu veriler karar ağacı, k-en yakın komşu ve rastgele orman makinası algoritmaları tarafından eğitilerek giriş durumuna göre çıkışların tahmin edildiği bir sistem tasarlanmıştır. Daha sonra tasarlanan sisteme, girişlerin durumu verilerek çıkışların tahmin etmesi beklenmiş ve bu çıkışlar sistemin gerçek çıkışlarıyla karşılaştırılmıştır. Tasarlanan sistemin akış şeması Şekil 6’daki gibidir.



Şekil 6. Tasarlanan sistemin akış şeması

3.1. Karar Ağacı Algoritması

Karar ağacı algoritması parametrik olmayan bir denetimli öğrenme algoritmasıdır. Her ne kadar regresyon algoritmalarında kullanılabilse de genellikle sınıflandırma algoritmalarında kullanılır (Charbuty, B., & Abdulazeez, A., 2021). Karar ağacı algoritması; Şekil 7' de görüldüğü gibi kök düğüm, dallar, iç düğümler ve yaprak düğümlerden oluşan hiyerarşik bir ağaç yapısına sahiptir. Bir karar ağacı, bir iç düğümün özelliği (veya özniteliği), dalın bir karar kuralını temsil ettiği ve her yaprak düğümün sonucu temsil ettiği akış şeması benzeri bir ağaç yapısıdır. Bir karar ağacındaki en üstteki düğüm, kök düğüm olarak bilinir. Bu kök düğüm öznitelik değeri temelinde bölünür. Bu akış şeması benzeri yapı, karar verilmesine yardımcı olur (IBM). Böylelikle insan seviyesindeki bir düşünceyi kolayca taklit edilebilen bir akış şeması diyagramı olarak görselleştirilebilir. Bu nedenle karar ağaçlarının anlaşılması ve yorumlanması kolaydır.



Şekil 7: Karar Ağacı Algoritması çalışması.

Karar ağacı algoritmasının bir çok çeşidi olsa da en çok kullanılanları; dallanmaları değerlendirmek için ölçüm olarak entropi ve bilgi kazancından yararlanan “Yinelemeli İkilemler 3” (ID3), karar ağaçlarındaki ayrık noktaları değerlendirmek için bilgi kazancını veya kazanç oranlarını kullanan (C4.5) ve “Sınıflandırma ve Regresyon Ağacı” (CART) algoritmalarıdır (Brodley, C. E., & Utgoff, P. E., 1995).

Karar ağacında veri kümesinin safsızlığını (impurity) ve rastgeleliğini (randomness) belirlemek önemlidir. Bu iki durumun belirlenmesi için Entropi kullanılır ve entropi Eşitlik 1’ de görüldüğü gibi hesaplanır.

$$\text{Entropi}(S) = - \sum_{c \in C} p(c) \log_2 p(c) \quad (1)$$

Bu eşitlikte S, entropinin hesaplandığı veri setini, c kümedeki sınıfları, p(c), c sınıfına ait veri noktalarının kümedeki toplam veri noktalarının sayısına oranını temsil eder. Entropi 0 ile 1 arasında değer alır.

S veri setindeki tüm örnekler bir sınıfa aitse, entropi sıfır olur. Benzer bir şekilde örneklerin yarısı bir sınıfta, diğer yarısı başka bir sınıftaysa, entropi en yüksek 1’de olacaktır. Ayrılacak en iyi özelliği seçmek ve en uygun karar ağacını bulmak için en az entropili öznelik kullanılmalıdır. Bilgi kazancı, belirli bir özellikteki bir bölünmeden önce ve sonra entropideki farkı temsil eder ve genellikle Eşitlik 2’ de görüldüğü gibi hesaplanır. Burada a, belirli bir özelliği veya sınıf etiketini temsil ederken, $|S_v| / |S|$ S_v’deki değerlerin veri kümesindeki değerlerin sayısına oranını temsil eder.

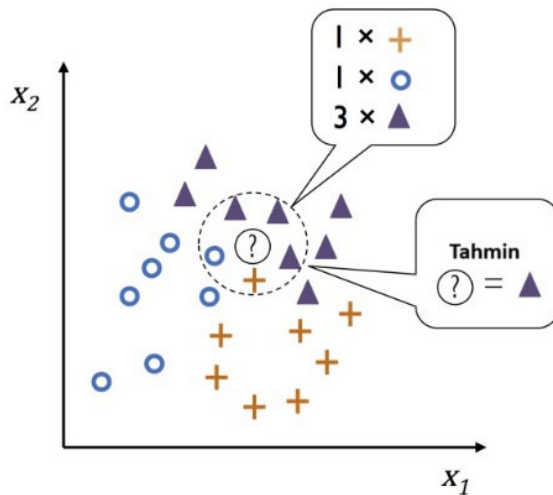
$$\text{Bilgi Kazancı}(S, a) = \text{Entropi}(S) - \sum_{v \in V} \frac{|S_v|}{|S|} \text{Entropi}(S_v) \quad (2)$$

Karar Ağacı, beyaz kutu tipi bir makine öğrenmesi algoritmasıdır (Hulstaert, L., 2019) Sinir Ağı gibi kara kutu tipi algoritmalarda bulunmayan dahili karar verme mantığını paylaşır. Eğitim süresi, sinir ağı algoritmasına göre daha hızlıdır. Karar ağaçlarının zaman karmaşıklığı, verilen verilerdeki kayıtların ve özneliklerin sayısının bir fonksiyonudur. Karar ağacı, olasılık dağılımı varsayımlarına bağlı olmayan, dağılımdan bağımsız veya parametrik olmayan bir yöntemdir. Karar ağaçları, yüksek boyutlu verileri iyi bir doğrulukla işleyebilir. Bu çalışmada, bahsi geçen avantajlarından dolayı karar ağacı algoritması kullanılmıştır.

3.2. K-En Yakın Komşu Algoritması

K-En yakın komşu sınıflandırıcı algoritması parametrik olmayan ve hem regresyon hemde sınıflandırma problemlerinde kullanılabilen bir algoritmadır. Sınıflandırmaları veya tahminleri yapmak için bireysel noktaların aralarındaki yakınlıklarını kullanır.

Sınıflandırma problemleri için “çoğunluk oyu” olarak adlandırılan bir sınıf etiketi atanır. Bu çoğunluk oyu belirli bir veri noktası çevresinde en sık temsil edilen etikettir. Birden fazla sınıf olduğunda; örneğin dört sınıf için bir sınıf hakkında bir sonuca varmak için mutlaka oyların %50’sine ihtiyaç yoktur; %25’in üzerinde oyu olan bir sınıf etiketi atanabilir. 3 farklı sınıf için 5 en yakın komşulu bir sistemin gösterimi Şekil 8’ de görüldüğü gibidir.



Şekil 8: 3 farklı sınıfta k=5 için knn algoritması

k-en yakın komşu algoritmasında amaç sınıf etiketi atayabilmek için, bir sorgu noktasının en yakın komşularını tanımaktır. Bunu yapabilmek için sorgu noktası ile diğer veri noktaları arasındaki mesafenin hesaplanması gerekir. Bu mesafe ölçümleri sorgu noktalarını farklı bölgelere ayıran karar sınırlarını oluşturur. Bir çift nokta arasındaki karar sınırı düz bir çizgi iken küresel düzeydeki en yakın komşu modellerinde dışbükey bir çok yüzlüdür. Mesafe hesaplanması için çok çeşitli yöntemler vardır. Bu yöntemlere öklid mesafesi, manhattan mesafesi, minkowski mesafesi örnek verilebilir.

3.2.1. *Öklid Mesafesi*: İki nokta arasındaki mesafe ölçümü için en çok kullanılan yöntemdir. Eşitlik 3 kullanılarak sorgu noktası ile ölçülen diğer nokta arasındaki düz bir çizgiyi ölçer.

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \quad (3)$$

3.2.2. *Manhattan Mesafesi*: Öklid mesafesinden sonra en popüler mesafe ölçüm yöntemlerinden birisidir. Taksi mesafesi veya şehir bloğu mesafesi olarak da adlandırılır. İki nokta arasındaki mutlak değeri ölçer (Eşitlik 4).

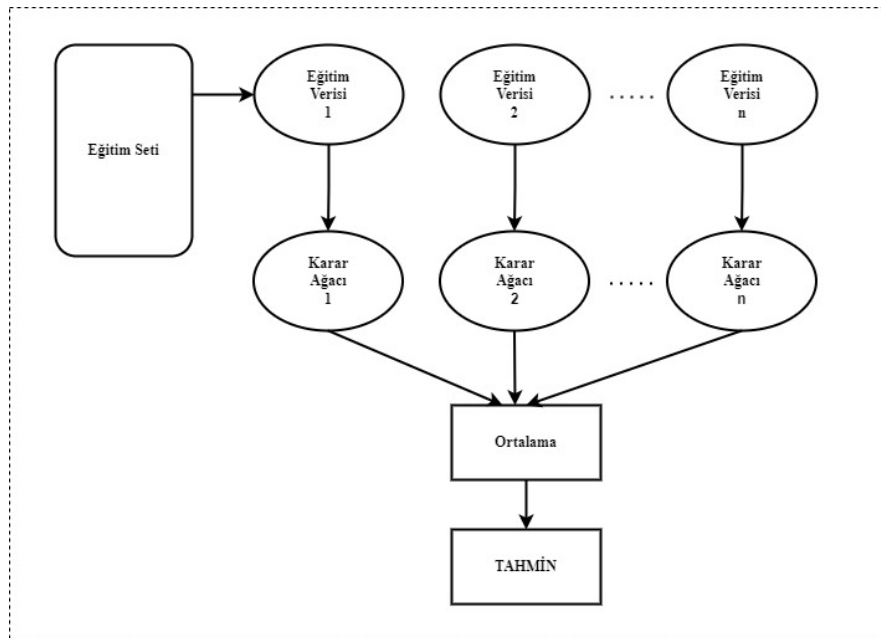
$$d(x, y) = \sqrt{\sum_{i=1}^m (|x_i - y_i|)} \quad (4)$$

Minkowski Mesafesi: Bu mesafe ölçüsü, Öklid ve Manhattan mesafe ölçümlerinin genelleştirilmiş şeklidir. Eşitlik 5'te yer alan p parametresi diğer mesafe metriklerinin oluşturulmasına olanak sağlar.

$$d(x, y) = (\sum_{i=1}^n |x_i - y_i|^p)^{\frac{1}{p}} \quad (5)$$

3.3. Rastgele Orman Algoritması

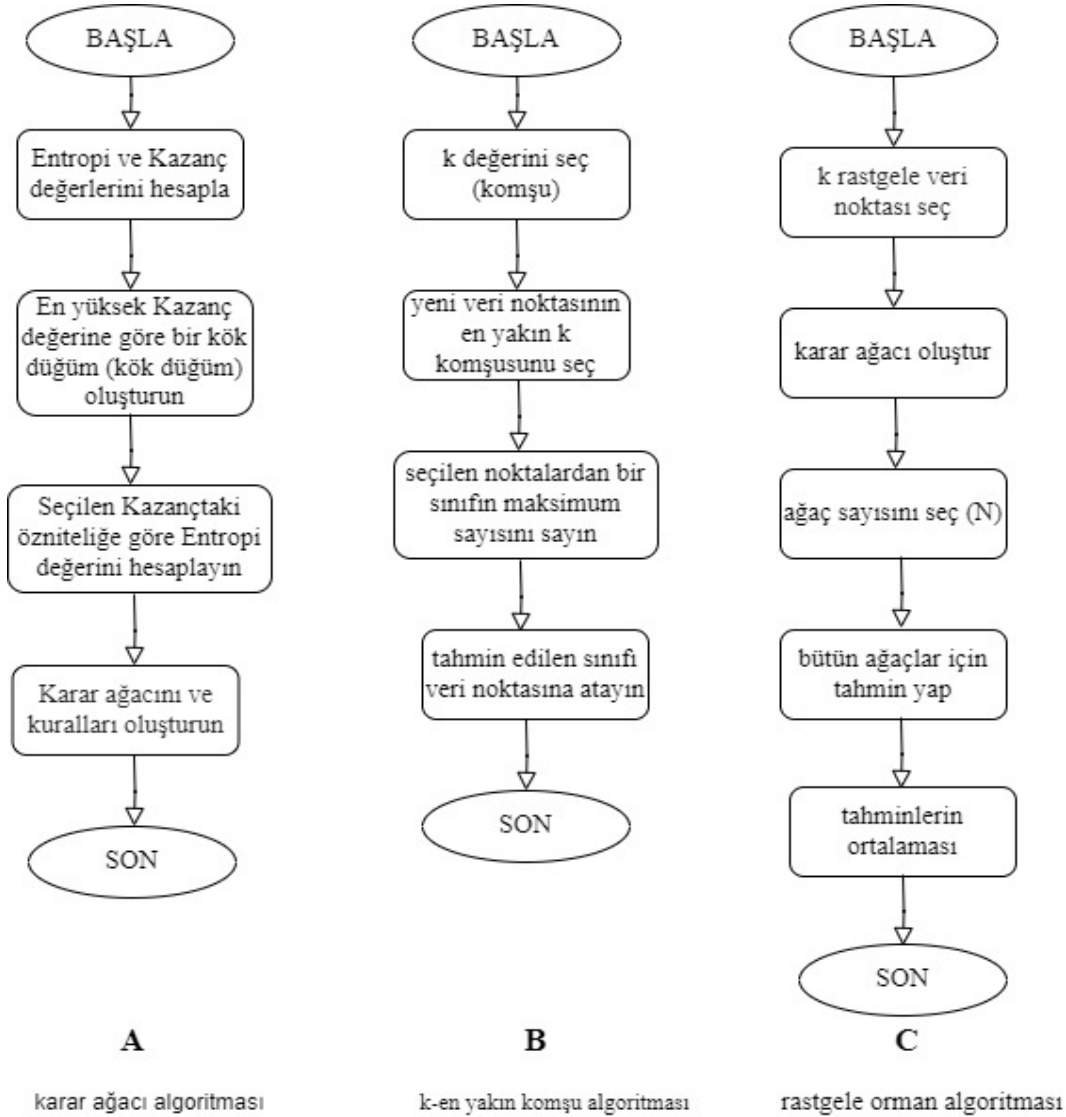
Rastgele orman algoritması, torbalama (bagging) yöntemi kullanan ve birden fazla sayıda karar ağacı algoritmasının birlikte çalışacak şekilde oluşturulduğu topluluk öğrenme yöntemidir. Bu algoritma, birden fazla olan bu karar ağaçlarının çıktılarını birleştirir ve daha doğru bir sonuç sunar. Rastgele orman algoritmasının çalışması Şekil 9'da görüldüğü gibidir.



Şekil 9: Rastgele Orman Algoritması çalışması.

3.4. Algoritmaların Akış Şemaları

Deney seti üzerinde bulunan PLC' nin 8 adet girişi ve 8 adet çıkışı olduğu için "çok sınıf-çok çıkış sınıflandırma" kullanılmıştır. Bu sınıflandırma yöntemine uygun oldukları için makine öğrenmesi algoritması olarak karar ağacı, k-en yakın komşu ve rastgele orman algoritmaları kullanılmıştır. Kullanılan bu makine öğrenmesi algoritmaları için akış şemaları Şekil 10' da görüldüğü gibidir.



Şekil 10: Kullanılan algoritmaların çalışma akış şemaları.

4. Bulgular

4.1. Giriş-Çıkış Kayıt Günlükleri

PLC ile tasarlanmış ve programı mevcut olan sisteme, 3.2. bölümünde detayları verilen yöntemle csv dosyası olarak giriş çıkış verileri alınmıştır. Makine öğrenmesi yöntemlerinde en önemli kısım veri olduğu için, mevcut sistem normal çalışma koşullarında çalışırken giriş çıkış verileri uzun süreli olarak kaydedilmiştir. Kaydedilen verilerin örneği Tablo 2 de görüldüğü gibidir.

Test düzeneği üzerinde sıcaklık, basınç ve titreşim gibi analog veri üreten analog sensörler bulunmaktadır. Analog sensörler bu çalışma kapsamında olmadığı için bu veriler kullanılmamıştır. Bu çalışmada sadece dijital giriş ve dijital çıkış verileri kullanılmıştır.

Tablo 2. Kaydı yapılan PLC giriş-çıkış veri örnekleri

Sıra No	I0.0	I0.1	I0.2	I0.3	I0.4	I0.5	I0.6	I0.7	I1.0	Q0.0	Q0.1	Q0.2
1	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
2	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
3	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
4	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
5	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0

4.2. Makine Öğrenmesi Algoritmalarının Çalıştırılması

Bu çalışmada kullanılmak üzere 7006 adet veri seti kaydedilmiştir. Bu veri seti üzerindeki Tablo 1’ de gösterilmiş olan PLC giriş verileri makine öğrenmesi algoritmalarının girişi, PLC nin çıkış verileri de seçilen makine öğrenmesi algoritmalarının çıkışı olarak belirlenmiştir. Bu giriş ve çıkış setlerine ek olarak test düzeneğinin çalışmaya başlama durumu da önemli olduğu için ‘zaman’ bilgisi de algoritmaya giriş olarak eklenmiştir.

7006 adet veri setinin 5600 adedi karar ağacı algoritmasını eğitmek için kullanılmış, geri kalan kısımda test verileri olarak kullanılmıştır. Bunun yanında makine öğrenmesinde en önemli durumlardan birisi veri setlerindeki bilgilerin kategorik ve nümerik olarak ayrılmasıdır. Bu çalışmada kullanılan veriler PLC nin dijital giriş çıkış verileri olduğu için herhangi bir kategorik veri yoktur. Ayrıca sistem üzerinden veriler anlık olarak kaydedildiği için herhangi bir eksik veri de bulunmamaktadır. Ayrıca giriş-çıkış verileri içerisinde herhangi bir eksik veri de bulunmamaktadır.

Çalışmada kullanılan algoritma Python programı ile yazılmış olup github linki eklenmiştir. Bu çalışmanın sonucu olarak ortaya çıkan doğruluk puanları karar ağacı, k-en yakın komşu ve rastgele orman algoritmaları için sırasıyla 0.914, 0.905 ve 0.948 çıkmıştır.

PLC ile tasarlanan sistemlerde pano üzerinde röleler kullanılır. PLC lerin işlem hızları milisaniyeler cinsinden olup (transistör çıkışlılar) röleler için bu süre 0,5 saniye civarındadır. Bazı durumlarda (özellikle asenkron motorların ani yön değiştirilmesi gibi) PLC ile rölelerin işlem hızlarının farklı olduğunda sistemlerde faz çakışmaları yaşanabilmektedir. Bu çalışmadaki kullanılan algoritmalar her ne kadar iyi çalışıyor olsa da anlık bazı çıkış değerleri problem oluşturabileceği göz önüne alınmalıdır. Test düzeneğindeki aynı anda aktif olduğunda mekanik probleme sebep olacak iki hareket belirlenip, bu hareketlerin aynı anda olmaması için röleler üzerinden önlem alınmıştır (kapalı kontaklarını açık kontakları üzerinden geçirerek). Bu adım sistemsal güvenliğin sağlanması için yapılmıştır.

5. Sonuç

Bu çalışmada, bilinmeyen bir PLC programının makine öğrenmesi tabanlı modellemesi yapılmıştır. Bu işlemin yapılabilmesi için ilk olarak bir test düzeneği kurulmuş, kurulan test düzeneğinin üzerinden Siemens web server aracılığıyla giriş çıkış veri günlüğü tutulmuştur. Alınan giriş verileri makine öğrenmesi algoritmasının giriş değişkenleri olarak atanmıştır. Bu giriş değerlerine ek olarak PLC programlarında işlemlerin oluş sırası önemli olduğu için zaman bilgisi de bir giriş olarak sisteme eklenmiştir. PLC nin çıkış verileri makine öğrenmesi algoritmasının çıkışları olarak atanmış ve algoritma çalıştırılmıştır. Makine öğrenmesi algoritması olarak karar ağacı, k-en yakın komşu ve rastgele orman sınıflandırıcı algoritmaları kullanılmış ve algoritmaların değerlendirilmesi için doğruluk puanı metriği ile yapılmıştır. Karar ağacı, k-en yakın komşu ve rastgele orman algoritmaları için doğruluk puanları sırasıyla 0.914, 0.905 ve 0.948 olarak elde edilmiş ve en iyi sonucu rastgele orman algoritması vermiştir.

Bu çalışma kapsamında PLC nin sadece dijital giriş ve çıkışları kullanılmıştır. Analog giriş ve çıkışlar kullanılmamıştır. Ayrıca PLC programı içerisinde zamanlayıcı-sayıcı gibi programlama ekipmanları da kullanılmamıştır. İlerleyen çalışmalarda analog sensör verileri ve zamanlayıcı- sayıcı gibi ekipmanlar kullanılarak da çalışmalar yapılabileceği öngörülmektedir.

Referanslar

- Barthelmey, A., Störkle, D., Kuhlenkötter, B., & Deuse, J. (2014). Cyber physical systems for life cycle continuous technical documentation of manufacturing facilities. *Procedia Cirp*, 17, 207-211.
- Bayindir, R., & Cetinceviz, Y. (2011). A water pumping control system with a programmable logic controller (PLC) and industrial wireless modules for industrial plants—An experimental setup. *ISA transactions*, 50(2), 321-328.
- Bolton, W. (2015). *Programmable logic controllers*. Newnes.
- Brodley, C. E., & Utgoff, P. E. (1995). Multivariate decision trees. *Machine learning*, 19(1), 45-77.
- Brusso, B. C. (2018). 50 Years of Industrial Automation [History]. *IEEE Industry Applications Magazine*, 24(4), 8-11.
- Carayol, A., & Nicaud, C. (2012). Distribution of the number of accessible states in a random deterministic automaton. In *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)* (Vol. 14, pp. 194-205). LIPIcs.
- Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01), 20-28.
- de MBA Dib, L., Fernandes, V., Filomeno, M. D. L., & Ribeiro, M. V. (2017). Hybrid PLC/wireless communication for smart grids and internet of things applications. *IEEE internet of things Journal*, 5(2), 655-667.
- Erickson, K. T. (1996). Programmable logic controllers. *IEEE potentials*, 15(1), 14-17.
- Glavaš, M., Krčmar, I., & Marić, P. (2021, March). Modelling of a sequential system. In *2021 20th International Symposium Infoteh-Jahorina (INFOTEH)* (pp. 1-6). IEEE.
- Guasch, A., Quevedo, J., & Milne, R. (2000). Fault diagnosis for gas turbines based on the control system. *Engineering Applications of Artificial Intelligence*, 13(4), 477-484.
- Hulstaert, L., (2019). Black-box vs. white-box models. <https://towardsdatascience.com/machine-learning-interpretability-techniques-662c723454f3>
- IBM, What is a Decision Tree?, <https://www.ibm.com/topics/decision-trees>.
- Kaid, H., Al-Ahmari, A., Nasr, E. A., Al-Shayea, A., Kamrani, A. K., Noman, M. A., & Mahmoud, H. A. (2020). Petri net model based on neural network for deadlock control and fault detection and treatment in automated manufacturing systems. *IEEE Access*, 8, 103219-103235.
- Lee, J. I., Chun, S. W., & Kang, S. J. (2002). Virtual prototyping of PLC-based embedded system using object model of target and behavior model by converting RLL-to-statechart directly. *Journal of systems architecture*, 48(1-3), 17-35.
- Liu, J., & Darabi, H. (2002, October). Ladder logic implementation of Ramadge-Wonham supervisory controller. In *Sixth International Workshop on Discrete Event Systems, 2002. Proceedings.* (pp. 383-389). IEEE.
- Lobov, A., Lastra, J. L. M., Tuokko, R., & Vyatkin, V. (2003). Methodology for modeling visual flowchart control programs using net condition/event systems formalism in distributed environments. In *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No. 03TH8696)* (Vol. 2, pp. 329-336). IEEE.
- Marzena Dobosz, (2020). The Role of PLCs in Industrial IoT. (<https://knowhow.distrelec.com/automation/the-role-of-plcs-in-industrial-iot/>).
- Otto, H. P., & Rath, G. (1996). State Diagrams A New Programming Method for Programmable Logic Controllers. In *Software Engineering for Manufacturing Systems* (pp. 27-37). Springer, Boston, MA.
- Özdem, Ö. C. (2016). Design of two experimental setups for programmable logic controller (PLC) laboratory. *International Journal of Electrical Engineering Education*, 53(4), 331-340.
- Rullan, A. (1997). Programmable logic controllers versus personal computers for process control. *Computers & industrial engineering*, 33(1-2), 421-424.

Silva, B. G., & De Sousa, M. (2016, September). Internal inconsistencies in the third edition of the IEC 61131-3 international standard. In 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1-4). IEEE.

Soliman, D., Thramboulidis, K., & Frey, G. (2012). Transformation of function block diagrams to Uppaal timed automata for the verification of safety applications. *Annual Reviews in Control*, 36(2), 338-345.

Sönmez, M., Nil, M., & Kandilli, U. İ. (2005). Üç serbest dereceli endüstriyel bir robotun yapay sinir ağları ile denetimi

Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., & Vasilakos, A. V. (2016). Software-defined industrial internet of things in the context of industry 4.0. *IEEE Sensors Journal*, 16(20), 7373-7380.

Zimmerman, G. P. (2008). Programmable logic controllers and ladder logic. Rapid City: Dr. Alfred R. Boysen, Department of Humanities, South Dakota School of Mines and Technology.