



**ÜRETİM SÜRECİNİN DOĞAL DİL
İŞLEME İLE MODELLENMESİ**

Yüksek Lisans Tezi

Hatice AY

Eskişehir 2019

ÜRETİM SÜRECİNİN DOĞAL DİL İŞLEME İLE MODELLENMESİ

Hatice AY

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Gürkan ÖZTÜRK

Eskişehir
Eskişehir Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Temmuz 2019

JÜRİ VE ENSTİTÜ ONAYI

Hatice Ay'ın “Üretim Sürecinin Doğal Dil İşleme İle Modellenmesi” başlıklı tezi 26/07/2019 tarihinde aşağıdaki jüri tarafından değerlendirilerek “Eskişehir Teknik Üniversitesi Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliği'nin ilgili maddeleri uyarınca, Endüstri Mühendisliği Anabilim dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Jüri Üyeleri

Unvanı Adı Soyadı

İmza

Üye (Tez Danışmanı) : Doç. Dr. Gürkan ÖZTÜRK

Üye : Prof. Dr. Onur KAYA

Üye : Dr. Öğr. Üyesi Sinan AYDIN

ÖZET

ÜRETİM SÜRECİNİN DOĞAL DİL İŞLEME İLE MODELLENMESİ

Hatice AY

Endüstri Mühendisliği Anabilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Temmuz 2019

Danışman: Doç. Dr. Gürkan ÖZTÜRK

Doğal dil işleme, yapay zeka ve dilbiliminin alt bilimi olarak yer almakta ve doğal dillerdeki metinlerin, ses dalgalarının algılanarak yazılımlarda çözümlenmesi ve bilgisayar ortamına aktarılması olarak tanımlanmaktadır. Doğal dil işlemede girdiler metin veya ses olabilir. Birçok alanda uygulanan doğal dil işlemenin bir uygulama alanı olan metin işleme ise, istenilen amaç doğrultusunda metne ait kelimeleri sınıflandırarak veya kümeleyerek analiz için işlenmeye hazır hale getirme işlemidir.

Gerçekleştirilen çalışmada doğal dil işleme ile üretim süreçlerinin modellenmesini sağlayacak bir program Python 3.7 kullanılarak yazılmıştır. Girdi olarak verilen metinde program metin işleme teknikleri ile metin üzerinde bir ön işleme gerçekleştirmektedir. Ardından üretimin hücre tipi, u tipi, atölye tipi veya seri üretim tiplerinden hangisi olduğunu belirlemekte ve buna uygun şekilde mevcut makineleri yerleştirip süreci modellemektedir. Daha sonra modellenen süreç kaydedilebilmekte/yazdırılabilmekte veya modellenen süreç üzerinden simülasyon aşamasına geçilebilmektedir. Simülasyon kısmında işlem sürelerine ait dağılım tipi ve parametre değerleri girilerek üretime ait çıktı miktarlarına ve makinelere ait kullanım oranlarına ulaşılmaktadır. Eğer veriye ait dağılım tipi bilinmiyorsa veri analizi kısmında verinin dağılım tipi Ki- kare ve Kolmogorov-Smirnov testleri kullanılarak belirlenebilmektedir.

Çalışmanın amacı, üretim süreçlerinin modellenmesi ve simülasyonun gerçekleştirilmesi için, başka bir programa ihtiyaç duyulmadan, sadece sesli anlatım ile tüm sonuçların elde edilmesini sağlayacak bir program için başlangıç niteliğinde bir temel oluşturmaktır.

Anahtar Sözcükler: Doğal dil işleme, Metin işleme, Süreç modelleme

ABSTRACT
MANUFACTURING PROCESS MODELING WITH NATURAL LANGUAGE
PROCESSING

Hatice AY

Department of Industrial Engineering

Eskişehir Technical University, Institute of Graduate Programs, July 2019

Supervisor: Assoc. Prof. Dr. Gürkan ÖZTÜRK

Natural language processing is a sub-science of artificial intelligence and linguistics and it is defined as the analysis of texts in natural languages, sound waves and their transfer to the computer environment. In natural language processing, the inputs can be text or sound. Text processing, which is an application area of natural language processing applied in many fields, is the process of getting ready for processing for analysis by classifying or clustering the words of the text for the desired purpose.

In this study, a program that will enable the modeling of natural language processing and production processes was written using Python 3.7. In the text given as input, the program performs a preprocessing on the text with text processing techniques. It then determines which type of production is cell type, u type, workshop type or mass production type, and accordingly locates existing machines and models the process. Then the modeled process can be saved/printed or the simulation process can be started over the modeled process. In the simulation section, by entering the distribution type and parameter values of the process times, the output amounts of the production and the usage rates of the machines are reached. If the distribution type of the data is unknown, the distribution type of data can be determined using Chi-square and Kolmogorov-Smirnov tests.

The study aims to provide an initial basis for a program that will provide all the results with only audio expression without the need for any other program for modeling and simulation of production processes.

Keywords: Natural language processing, Text processing, Process modeling

TEŞEKKÜR

Bu çalışma süresince, değerli bilgi ve tecrübelerini benimle paylaşan, her ihtiyaç duyduğumda bana vakit ayıran ve beni destekleyen sayın danışman hocam Doç. Dr. Gürkan ÖZTÜRK'e teşekkürlerimi sunarım.

Ayrıca hayatımın her anında yanımda olan ve bu çalışma zarfında beni destekleyen ailem ve kıymetli dostlarım Tuğçe YAVUZ, Tunahan ÇINAR, Özge ALÇAY ve Mustafa GÜNER'e teşekkür ederim.



ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmamın Eskişehir Teknik Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı”yla tarandığını ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Hatice AY

İÇİNDEKİLER

	<u>Sayfa</u>
BAŞLIK SAYFASI	i
JÜRİ VE ENSTİTÜ ONAYI.....	ii
ÖZET	iii
ABSTRACT.....	iv
TEŞEKKÜR	v
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	vi
İÇİNDEKİLER	vii
ŞEKİLLER TABLOSU	ix
1. GİRİŞ	1
2. LİTERATÜR TARAMASI.....	4
3. PROBLEM TANIMI.....	8
4. YÖNTEMLER	10
4.1. Doğal Dil İşleme	10
4.1.1. Metin işleme.....	12
4.2. Üretim Süreci	13
4.2.1. Seri üretim	13
4.2.2. Atölye tipi üretim	14
4.2.3. Hücre tipi üretim	15
4.2.4. U Tipi Hatlar	16
4.3. Simülasyon	17
4.3.1. Olasılık dağılımları	19
4.4. Kullanılan Kütüphaneler	22
5. UYGULAMA	24
5.1. Süreç Modelleme	26
5.2. Simülasyon.....	38

	<u>Sayfa</u>
5.3. Veri Analizi.....	42
6. SONUÇ VE ÖNERİLER.....	46
KAYNAKÇA.....	48
ÖZGEÇMİŞ	



ŞEKİLLER TABLOSU

Sayfa

Şekil 3.1. Bir cnc- torna makinesine ait veriler	9
Şekil 4.1. N basamaklı seri üretim hattı.....	14
Şekil 4.2. Atölye tipi üretim süreci	15
Şekil 4.3. Hücre tipi üretim süreci	16
Şekil 4.4. U tipi hatlar.....	17
Şekil 4.5. Simülasyon adımlarına ait akış şeması.....	18
Şekil 5.1. Programa ait akış şeması	25
Şekil 5.2. Süreç modelleme algoritmasına ait ilk 5 adım	26
Şekil 5.3. Süreç modelleme algoritmasına ait 6. adım	26
Şekil 5.4. Süreç modelleme algoritmasına ait 7. adım	28
Şekil 5.5. Süreç modelleme algoritmasına ait 8. adım	29
Şekil 5.6. Süreç modelleme algoritmasına ait 9. adım	30
Şekil 5.7. Süreç modelleme algoritmasına ait 9. adımın devamı	31
Şekil 5.8. Süreç modelleme algoritmasına ait 10. adım	32
Şekil 5.9. Ana sayfa ara yüzü	33
Şekil 5.10. Metin giriniz uyarısı	33
Şekil 5.11. Makine ekleme alanı	34
Şekil 5.12. Makine ekleme uyarısı	34
Şekil 5. 13. Metin 1 ile elde edilen model	35

Şekil 5.14. Metin 2 ile elde edilen model	36
Şekil 5.15. Metin 3 ile elde edilen model	37
Şekil 5.16. Metin 4 ile elde edilen model	37
Şekil 5.17. Simülasyon modelline ait algoritma adımları	38
Şekil 5.18. Metin 1 ile elde edilen simülasyon ara yüzü	39
Şekil 5.19. Metin 2 modeli ile elde simülasyon ara yüzü	40
Şekil 5.20. Metin 3 modeli ile elde edilen simülasyon ara yüzü	41
Şekil 5.21. Metin 1 ile elde edilen modelin simülasyon çıktısı	42
Şekil 5.22. Metin 2 ile elde edilen modelin simülasyon çıktısı	42
Şekil 5.23. Veri analizi algoritmasına ait adımlar	43
Şekil 5.24. Veri seti analiz sonuçları	44
Şekil 5.25. Programda kullanılabilecek diğer araçlar	44

1. GİRİŞ

İnsan davranışı ve hareketlerinin makinelere aktarılması olarak tanımlanabilen yapay zeka kavramı ilk olarak 1956 yılında gerçekleştirilen bir konferansta ortaya atılmıştır. Uzun süredir üzerinde çalışmalar yapılan yapay zeka günümüzde; ulaşım, sağlık, iletişim, tekstil, finans, eğitim, pazarlama ve savunma sanayii gibi birçok alanda geliştirilerek yaşamın hemen hemen her noktasında bir şekilde kullanılmaktadır. Bunun yanı sıra artık yapay zeka ile geliştirilen birçok uygulama ile insan-makine etkileşimi gerçekleşmektedir. İnsan-makine etkileşiminin sağlanabilmesi için kullanılan doğal dillerin bilgisayarlara öğretilmesine ihtiyaç duyulmuştur. Bu durum sonucunda da dilbilim ve yapay zekanın bir alt dalı olan doğal dil işleme doğmuştur.

Dilbilimi, dilin genel ve özel niteliklerini ve dil olaylarını inceleyen, dillerin doğuşlarını zaman içindeki gelişmelerini, yeryüzündeki yayılışlarını ve aralarındaki ilişkileri araştıran bilim dalıdır. Doğal dil işleme ise, yapay zeka ve dilbiliminin alt bilimi olarak yer almakta ve doğal dillerdeki metinlerin, ses dalgalarının algılanarak yazılımlar aracılığıyla çözümlenmesi ve bilgisayar ortamına aktarılması olarak tanımlanmaktadır (http-1 ve http-2).

Gelişen teknoloji ile makine-insan iletişiminin artması, insanın makineden olan beklentilerini yükseltmektedir. İnsanların makineler tarafından anlaşılmayı beklemesi, kullanımı kolay makinelerin ve uygulamaların doğmasına neden olmaktadır. Bu sayede insanın yaşam kalitesinin artması ile akıllı makinelerin her alanda kullanılması kaçınılmaz hale gelecektir. Bu önermeyi destekler nitelikte ve yakın dönemde doğal dil işleme kullanılarak gerçekleştirilen birçok uygulama mevcuttur. Örnek olarak; Apple markasına ait sesli asistan uygulaması olan Siri, Outlook ve Gmail gibi programlarda mesajların gruplanması veya spam olan mesajların tespit edilmesi, arama motorlarında yazılan metnin algılanması, Microsoft Word programında metin hatalarının gösterilmesi veya metindeki herhangi bir kelimenin istenen bir kelimeyle yer değiştirilmesi, yaygın olarak bankacılık ve iletişim sektörlerinde kullanılan müşteri hizmetleri servisinin sesli komutlarla yönlendirilmesi verilebilir.

Literatürde insan- makine etkileşimini sağlayan doğal dil işleme ile ilgili mevcut çalışmalar incelendiğinde birçok alanda uygulama bulunmasına rağmen süreç modellemesine ilişkin çok az sayıda çalışmaya rastlanmıştır. Doğal dil işlemenin süreç modellemesi için verimli ve kullanışlı olması ayrıca literatürde bu alanda yapılmış pek fazla çalışma olmaması nedeniyle; bu çalışmada doğal dil işleme kullanılarak üretim

süreçlerini modelleyen bir yazılım geliştirilmiştir. Farklı üretim tiplerine ait süreçleri modelleyen ve süreçlerin simülasyonlarını gerçekleştiren bu yazılım ile sesli anlatım kullanılarak sürece ait ihtiyaç duyulan tüm sonuçları verebilecek bir program için temel oluşturulması amaçlanmıştır. Özetle, bu çalışma üretim süreçlerinin modellenmesi ve süreç simülasyonunun gerçekleştirilmesi için insan ve makine arasında sesli anlatıma dayanan bir iletişim süreci oluşturmada başlangıç niteliği taşımaktadır.

Doğal dil işleme girdi biçimine göre metin ve ses işleme olarak ikiye ayrılmaktadır. Literatürde ses işleme üzerine çok sayıda çalışma bulunmamaktadır. Ses üzerine olan mevcut çalışmalarda ise genellikle sesli anlatımların yazılı bir metne dönüştürülerek metin işleme olarak ele alınmıştır. Günümüzde birçok alanda kullanılan doğal dil işlemenin uygulama alanlarından bazıları; soru cevaplama, otomatik tercüme, bilginin getirilmesi, kelime işleme, veri taban ara yüzü, internet arama motorları, metin özetleme, argümanların birleştirilmesi ve metin işleme olarak sıralanabilmektedir (Şeker, 2015).

Gerçekleştirilen çalışma metin işleme alanına ait bir uygulamadır. Metin işleme, bir metin üzerinde yapılan kategorize etme, bölümlendirme ve kümeleme işlemi olarak tanımlanmaktadır. Belirlenen amaçlara göre metnin farklı şekilde işlenmesi gerekmekte ve bu işleme biçimi her amaca uymamaktadır. Birçok tekniği bulunan metin işlemenin, her tekniğinin her uygulamada kullanılması gerekmemektedir. Metin işleme teknikleri ile incelenecek metin, amaçlar doğrultusunda uygun hale getirilerek istenilen düzende bir metin oluşturulmakta ve bu oluşturulan metin üzerinden işlemler devam etmektedir.

Çalışma kapsamında, bir üretim sürecini anlatan metni işleyerek, sürecin modellenmesini sağlayan bir program yazılmıştır. Program yazım dili olarak Python 3.7. kullanılmıştır.

İlk aşama olarak programa verilen metin, metin işleme teknikleri ile istenilen hale getirildikten sonra işlenmiş metin üzerinden sürecin seri, hücreli, U tipi ve atölye tipi üretim süreçlerinden hangisine uyduğu belirlenmekte, daha sonra kullanılan makineler tespit edilip belirlenen üretim sürecine göre yerleştirilerek süreç modellenmektedir. Model tekrar kullanılmak üzere kaydedilebilmekte ve çeşitli formatlarda çıktı alınabilmektedir. Elde edilen model üzerinden simülasyon çalışmaları yapılabilmektedir. Oluşturulan süreçteki makinelerle ilgili sıralamalar simülasyon aşamasına aktarılmakta ve kullanıcı tarafından girilen makine işlem sürelerine ait dağılım tipi ve parametreler ile üretim süreci için çıktılar elde edilmektedir. Elde edilen çıktılar gerçekleştirilen üretim miktarı, ara stok miktarları, parça bekleme süreleri ve makine kullanım oranlarından

oluřmaktadır. Eęer simölasyon ařamasında kullanılacak verilere ait daęılım tipi bilinmiyorsa veri analizi kısmından daęılım tipi ve parametre deęerleri hesaplanmaktadır. Daęılım tipleri Ki- Kare ve Kolmogorov- Smirnov uygunluk testleri kullanılarak belirlenmektedir. Her ařamadaki ıktılar kaydedilebilmekte ve yazdırılabilmektedir.

alıřmanın devamında doęal dil iřleme ve metin iřleme ile ilgili alıřmaları kapsayan bir literatür taraması ve daha sonra problem tanımı yapılmıřtır. Ardından kullanılan yöntemler, uygulama ve sonular kısmına yer verilmiřtir.



2. LİTERATÜR TARAMASI

50 yılı aşkın bir süredir üzerinde çalışma yapılan doğal dil işleme, doğal dillerdeki metinlerin, ses dalgalarının algılanarak yazılımlar aracılığıyla çözümlenmesi ve bilgisayar ortamına aktarılmasıdır. Günümüzde doğal dil işleme sağlık, hizmet, iletişim, sanat, eğitim gibi hemen hemen her alanda kullanılmakta ve her geçen gün insan hayatındaki rolü artmaktadır.

Doğal dil işleme alanındaki ilk çalışmalardan biri Woods (1970) tarafından yapılmıştır ve bu çalışma sayesinde dilin matematiksel modeli oluşturulmuştur. Çalışmada doğal dil analizi için genişletilmiş geçiş ağları yaklaşımı önerilmiştir. Bu yaklaşım ile sonlu sayıdaki bir küme kullanılarak verilen her bir harf için bir sonraki harfin ne olabileceği sıralanmaktadır. Dempster vd. (1977) yaptıkları çalışmada eksik, kısaltılmış ve sansürlenmiş veri setinden doğru bilgi çekmeyi sağlayan bir algoritma geliştirmişlerdir. Bahl vd. (1983) tarafından gerçekleştirilen çalışmada konuşma tanıma üzerine istatistiksel algoritmalar önerilmiştir. Bu algoritmalar ile konuşma girdisi analiz edilerek eksik veri tamamlanmaya çalışılmıştır. Brown vd. (1990) tarafından yapılan çalışmada istatistiksel yaklaşımlar kullanılarak oluşturulan model ile Fransızcadan İngilizceye çeviri amaçlanmıştır. Hindle ve Rooth (1993) yaptıkları çalışmada kişinin belirsiz olduğu edat cümlelerinde kişiyi tahmin etmeye çalışan bir algoritma geliştirmişlerdir. Nivre ve Scholz (2004) ise İngilizce cümleleri öğelerine ayıran bir algoritma tasarlamışlardır. Bu algoritmanın en büyük özelliği hafıza tabanlı öğrenmeye dayalı olmasıdır.

Önerilen birçok algoritma ve yaklaşımlarla gelişen doğal dil işleme, günümüzde karşılaşılan problemleri çözmek veya çözüm aşamasını kolaylaştırmak için kullanılmaktadır. Literatürde birçok sektörde bu alanda yapılmış çalışmalar mevcuttur.

Tan vd. (2018) tarafından doğal dil işleme ile bel ağrılarına yönelik x-ray ve MR raporlarına ait sonuçların sınıflandırılmasını amaçlayan açık kaynak kodlu bir model geliştirilmiştir. Makine öğrenme ve kural tabanlı olmak üzere modelde iki algoritma kullanılmıştır. Hasta raporları üzerinde uygulanan iki algoritmanın da %90 üzerinde doğruluk göstermesiyle beraber en iyi sonuç makine öğrenmeli model ile elde edilmiştir. Benzer bir çalışma Chen vd. (2019) tarafından hepatosellüler kanser hastalığının seviyesinin belirlenmesi için anlamlı bilgi çıkarmakta kullanılmak amacıyla doğal dil işleme ile hibrit ve kural tabanlı iki algoritma geliştirilmiştir. Çin'deki hastalara ait raporlar kullanılarak test edilen algoritmalarda daha iyi sonucu %80 üzerinde bir

başarıyla kural tabanlı algoritma vermiştir. Geliştirilen algoritma ile manuel sonuçlar karşılaştırıldığında algoritmanın daha başarılı olduğu gözlemlenmiştir. Sağlık alanında gerçekleştirilen bir diğer çalışma Swartz vd. (2017) tarafından radyoloji alanında yapılmıştır. Hasta raporları incelenerek pozitif ve negatif olarak sınıflandırılması amaçlanan çalışmada, doğal dil işleme ile kullanıcı tarafından kişiselleştirilebilen açık kaynak kodlu bir araç oluşturulmuştur. Bu araç kullanılarak oluşturulan sınıflandırmada %95 üzerinde bir doğruluk oranına ulaşılmıştır.

Zhang vd. (2019) tarafından şantiye kaza raporları incelenerek kaza nedenlerini doğal dil işleme ile belirleyen bir model tasarlanmıştır. Bu modelde raporlarda yer alan kelimeler analiz edilerek kaza nedenleri belirlenmeye çalışılmıştır. Algoritma işleyişinde yalın bayes, doğrusal regresyon, K- en yakın komşu, karar ağacı, destek vektör makinesi ve ensemble yöntemi olmak üzere 5 farklı algoritma kullanılmıştır. 16323 raporun incelendiği modelde en fazla doğru kaza nedenini ensemble yöntemi ile oluşturulan doğal dil işleme modeli tespit etmiştir.

Matci ve Avcan (2018) tarafından gerçekleştirilen çalışmada, Eskişehir’de bulunan okul adreslerine ilişkin en doğru koordinatı veren adres biçimi oluşturulmaya çalışılmıştır. Düzenlenmeden önceki adresler ile düzenlendikten sonraki adresler Google geocoding Api ve ArcGIS Api sistemlerine girilerek istenilen adrese göre verilen koordinatların doğruluğu karşılaştırılmıştır. Adresler doğal dil işleme kullanılarak PTT formatında düzenlenmiştir. Her iki sistemde de düzeltilen adresler önceki yazım şekline göre yüksek bir doğruluk oranı göstermiştir.

Wachtel vd. (2016) tarafından doğal dil işleme ile MS Excel’de fonksiyon oluşturabilmek için doğal dil işleme ile metin girdisi fonksiyona dönüştürülmüştür. Girdi olan cümledeki her kelime matematiksel işlev, sayı, hücre konumu, belirtme gibi işlevine göre gruplandırılmış daha sonra belirlenen gruplar çekilerek anlamlarına göre işlem gerçekleştirilmiştir. Oluşturulan model, girdideki kelimeleri doğru grupta %82 başarı elde ederken doğru fonksiyon oluşturmada %79,5 başarı elde etmiştir.

Manns vd. (2015) tarafından yeni bir ürüne ait montaj hattı oluşturulması için doğal dil işleme ve veri madenciliği kullanılarak iki model oluşturulmuştur. Daha önceki ürünlere ait veri setleri ile eğitilen modeller ile yeni ürünün üretiminde kullanılacak montaj hattının oluşturulması sağlanmıştır. Elde edilen sonuçlar karşılaştırıldığında doğal dil işleme yöntemi ile oluşturulan modelin veri madenciliği ile oluşturulan modele göre daha iyi sonuç verdiği görülmüştür.

Tanguy vd. (2015) tarafından havayolu güvenlik raporları için bir doğal dil işleme modeli oluşturulmuştur. Bu çalışmada, farklı tip ve dilde yazılan raporlar içerisinde acil güvenlik sorunu teşkil eden durumların doğal dil işleme kullanılarak tespit edilmesi amaçlanmıştır. Tekrar eden veya bilinen durumların tespitinde elde edilen başarı oranı yüksek iken nadir karşılaşılan ama acil güvenlik sorunu teşkil eden durumların tespitindeki başarı oranı oldukça düşüktür.

Zou vd. (2017) tarafından şantiye kazalarına ait veriler kullanılarak risk yönetimine ilişkin analiz yapılması amaçlanmıştır. Çalışmanın gerçekleştirildiği firmada kaza raporları geniş bir veri seti oluşturduğundan veri analizi doğal dil işleme kullanılarak yapılmıştır. Kaza raporu metinleri işlenerek en çok karşılaşılan 10 kaza sebebi sıralanmıştır. Elde edilen veriler risk yönetiminde kullanılmıştır. Bu sayede risk analizinde kısa sürede fazla sayıda kaza raporu incelenerek daha verimli sonuçlara ulaşılmıştır.

Cheng vd. (2014) tarafından doğal dil işleme ile verilen komutlarda oluşabilecek belirsizliklerin önüne geçmek amacıyla yapılan bu çalışmada robotik bir kola konuşarak komut verilmekte ve komutlar robot tarafından görev sıralamasına konmaktadır. Eğer yeni görev mevcut görev ile uyumsuzluk yaratıyor ise robot tarafından ara görevler oluşturularak geçiş sağlanmaktadır. Robotik alanda yapılan bir diğer çalışmada Wächter vd. (2018) tarafından insansı bir robotun sesli komutları algılayarak verilen görevi yerine getirmesi amaçlanmıştır. Bir önceki çalışmanın daha ileri seviyesi olan bu çalışmada doğal dil işleme kullanılarak programlanan robota bir mutfak alanı öğretilmiştir. Verilen komut ile robot tarafından istenilen eylemlerin planlaması yapılmaktadır. Bu planlar doğrultusunda verilen komutu yerine getirmeye çalışan robot başarısızlık durumunda yeni bir plan ile tekrar başlamaktadır. Oluşturulan örnek bir senaryo ile tüm komutlar robot tarafından yerine getirilebilmiştir.

Van Le vd. (2018) tarafından gerçekleştirilen çalışmada sağlık raporlarına ait sonuçlar makine öğrenmesi ile incelenerek kendine veya başkasına zarar verebilecek kişilerin tespiti amaçlanmıştır. Serbest metin şeklinde yazılan raporlar doğal dil işleme ile işlenmiş ardından 7 farklı makine öğrenme algoritması kullanılarak ruh hastalıklarının belirtilerine dair 4 farklı sözlükle karşılaştırılmıştır. Kendine veya başkasına zarar veren kişilerin sonuçlarının da bulunduğu raporlarda, yazılan kelimeler sözlük anlamları ile bütünleştirilmiş, lojistik karar ağacı ve destek vektör makinesi yöntemleri ile oluşturulan modeller kullanılarak bu kişiler %50'nin üzerinde doğruluk ile tespit edilmiştir. Doğal dil

işleme ile oluşturulan bu modeller sayesinde sağlık raporları analiz edilerek kendine veya başkasına zarar verebilecek kişiler önceden belirlenebilmektedir.

Large vd. (2017) tarafından yapılan çalışmada, bir otomobil simülasyonunda sürüş asistanını kullanarak 25 farklı kişiyle yapılan 10'ar dakikalık deneylerde, insanların dijital sürüş asistanıyla olan iletişim biçimi doğal dil işleme ile kategorize edilmeye çalışılmıştır. Deney sonucunda insanların dijital sürüş asistanıyla gerçek bir insanmış gibi iletişim kurdukları gözlemlenmiştir. Deneklerin o anki duygu durumuna göre konuşma üslupları değişkenlik göstermiş ve üslup şekilleri doğal dil işleme ile 7 kategoriye ayrılmıştır. Çalışma sonunda yazarlar insanla iletişime geçecek asistan uygulamalarında bu durumun göz önüne alınması gerektiğini belirtmiştir.

Wichmann vd. (2018) tarafından yapılan bir çalışmada Toyota'ya ait tedarik zinciri doğal dil işleme ile görselleştirilmiştir. Uygulama metin alma, ön işleme, ayrıştırma, bilgi çıkarma, tedarik zinciri haritalama ve görselleştirme adımlarını gerçekleştirerek girilen veriden tedarik zincirini elde etmiştir. Elde edilen sonuçla tedarik zinciri oluşturulabileceği, mevcut zincirin doğrulanabileceği veya mevcut zincire müşteri eklenebileceği gösterilmiştir.

Milkov vd. (2013) tarafından yapılan çalışma ile bir matematiksel model oluşturularak büyük veri seti içerisindeki benzer kelimelerin eşleşmesi sağlanmıştır. Bu çalışmada farklı olarak, eşleştirilen kelimelerin anlam ifadeleri öncelik olarak kullanılmıştır. Örneğin; Paris ve Fransa'nın, Berlin ve Almanya'nın eşleştirilmesi gibi.

Literatürdeki doğal dil işleme ile ilgili çalışmalar incelendiğinde genellikle metin işleme üzerinde çalışmalar yapıldığı görülmüştür. Robotik çalışmalarda kullanılan ses girdili olan uygulamalarda da sesli girdi yazılı metne dönüştürülerek işlemler yapılmaktadır. Ayrıca incelenen çalışmalar ile doğal dil işlemenin birçok sektör ve alanda uygulandığı görülmüştür. Buna rağmen üretim süreçlerinin oluşturulmasında yapılan çalışma sayısı oldukça azdır. Manns vd. (2015) tarafından montaj hattı modelinin tahmini üzerinde yapılan çalışma üretim süreçlerinin doğal dil işleme ile oluşturulmasına örnek olarak gösterilebilir. Bu çalışmada mevcut veriler incelenerek bir ürüne ait montaj hattı tahmin edilmeye çalışılmıştır. Mevcut çalışmanın bu çalışmadan farkı ise tahmin üzerine olmaması, kullanıcı tarafından anlatılan sürecin metin işlenerek modellenmesi ve oluşturulan sürece dair bir simülasyonun da yapılabilmesidir. Bu çalışma insan- makine etkileşimi içeren bir sistem oluşturmaktadır.

3. PROBLEM TANIMI

En basit haliyle süreç, bir işin tamamlanması için gerçekleştirilen tüm işlem basamakları olarak tanımlanabilir. İş hayatında süreç kavramı karşımıza daha çok “iş süreci” olarak çıkmaktadır. Süreç yönetimi ise, bir işletmenin performansını süreçleri temel alarak sürekli iyileştirmesini sağlayan yönetim biçimidir. Bu kavramlar Toyota firmasının “Toplam Kalite Yönetimi” felsefesini yaratmasıyla doğmuştur. 1980’li yıllarda Japonlar tarafından bu felsefe benimsenmiş ve dünya pazarında büyük bir başarı elde etmişlerdir. Sürecin en doğru şekilde oluşturulup analiz edilmesi ve sürecin daima iyileştirilmeye çalışılması bu başarının en büyük faktörlerindendir. Hammadde tedariginden müşteriye kadar giden her adım kendi içinde bir iş sürecini oluşturmaktadır. Bu çalışmada iş süreçlerinden biri olan üretim süreçleri ele alınmıştır.

Üretim süreci, hammadde veya yarı mamullerin, üretim faktörleri aracılığıyla farklı bir çıktıya dönüştürülme sürecidir. Üretim süreçlerinin en hızlı ve doğru şekilde tanımlanması ve analiz edilmesi sayesinde üretimde yapılan yanlışlar tespit edilebilmekte ve bu sayede verimin arttırılmasını sağlayacak çalışmalar yapılabilmektedir. Üretim sürecinin modellenmesindeki başarı arttıkça, üretimin takibinin, iş yükü dengelemenin ve süreç takibinin (makinelerin performansı, dar boğaz noktalarının tespiti, üretimdeki kayıpların gözlemlenmesi vb.) de başarısı artmaktadır. Bu nedenle rekabetin yüksek olduğu üretim sektöründe firmalar için üretim sürecinin modellenmesi önem arz etmektedir. Gelişen teknoloji ile yapay zekanın her sektörde kullanılmaya başlanması insan-makine etkileşiminin artmasına yol açmıştır. Bu sayede gelişen doğal dil işleme ile insan hayatında karşılaşılan problemler hızlı bir şekilde çözülebilmektedir. Bu çalışmanın çıkış noktası doğal dil işleme ile üretim süreçlerinin modellenmesini ve oluşturulan sürecin analizi için simülasyon modelinin oluşturulmasını sağlayan bir programın tasarlanmasıdır. Yola çıkılan bu tasarıda tüm veri, sesli anlatımı algılayan yazılım tarafından çekilerek süreç modeli çizilip simülasyon oluşturulabilmekte ve çıktılar bilgisayar tarafından sesli ve yazılı olarak aktarılabilmektedir. Ayrıca simülasyon aşaması bir adım daha öteye götürülerek, anlatımda makine ismi ve modeli belirtilip daha sonra program tarafından makineye dair verilerin internet ortamından çekilebilmesi hedeflenmektedir. Örneğin; Şekil 3.1’deki X firmasının Y modelinin cnc-torna makinesine ait verileri çekilip makine görseli süreç modellenmesinde, gerekli teknik veriler ise simülasyon aşamasında veya süreçte kullanılan makinelere dair bilgi depolamada kullanılabilir. Bu tasarı sayesinde insan- makine etkileşimi ile başka bir

programa ihtiya duyulmadan sre ve simlasyon modeli hızlı bir şekilde gerekleřtirilebilecektir.

	Teknik zellikler	
	Maks. evirme apı	: $\varnothing 650$ mm
	Maks. Tornalama apı	: $\varnothing 500$ mm
	Maks. Tornalama Boyu	: 750 mm
	ubuk Kapasitesi	: $\varnothing 75$ mm
	Maks. İřmili Hızı	: 1500 dev./dak.
	İřmili Normu	: A2-8
	İřmili Delik apı	: $\varnothing 91$ mm
	İřmili Konięi	: 100
	İřmili Merkezinin Yerden Ykseklięi	: 1065 mm
	Punta Pinl apı	: $\varnothing 130$ mm
	Punta Pinl Hareketi	: 100 mm
	Punta Pinl	: 5#

řekil 3.1. Bir cnc- torna makinesine ait veriler

alıřma kapsamında, tasarlanan bu model iin bařlangı nitelięinde bir temel oluřturmak amacıyla bir program yazılmıřtır. Bu ama doęrultusunda ilk nce metin verisinin grselleřtirilmesi gerekmektedir. Bunun iin ncelikle girilen metin, metin iřleme teknikleri ile dzenlenmeli ardından ama doęrultusunda gerekli bilgiler ıkarılmalıdır. İhtiya duyulan ilk bilgi verinin hangi retim sreci modeline uyduęu bilgisidir. Daha sonra ise retim srecinde kullanılan makine isimleri belirlenmeli, belirlenen makinelerin retim sreci modelinde nereye ve nasıl yerleřtirileceęi tasarlanarak sre grselleřtirilmelidir. Modellenen sre zerinden ise veriler simlasyon ařamasına aktarılmalı ve ihtiya duyulan dięer veriler kullanıcı tarafından girilerek simlasyon ařaması gerekleřtirilmelidir.

Ama doęrultusunda belirlenen problem doęal dil iřleme kullanılarak Python 3.7 ile kodlanarak zlmř ve sre modellenmiřtir.

4. YÖNTEMLER

Bu başlık altında çalışmada kullanılan doğal dil işleme, üretim süreçleri ve simülasyon yöntemlerine ve programın yazım dili olan Python'da kullanılan kütüphanelere yer verilmiştir.

4.1. Doğal Dil İşleme

Yapay zeka ve dilbiliminin bir alt bilimi olan doğal dil işleme; doğal dillerdeki metinlerin, ses dalgalarının bilgisayar tarafından algılanarak yazılımlar aracılığıyla çözümlenmesi ve bilgisayar ortamına aktarılması olarak tanımlanmaktadır. 50 yıldan fazla bir süredir üzerinde çalışma yapılan doğal dil işleme, bilgisayarlarla etkileşime geçmek için gereklidir. Yazılı metin ve ses olarak 2 girdi tipi mevcuttur. Yapılan çalışmalarda genellikle yazılı metin girdisi kullanılmaktadır. Ses girdilerine dair yapılan çalışmaların çoğunluğu, ses girdisinin yazılı metne dönüştürülüp işlenmesi şeklindedir.

Uzun süredir doğal dil işleme üzerinde çalışmalar yapılmasına rağmen hala karşılaşılan birçok zorluğu vardır. Bu zorluklar; kuralsız ve anlaşılmaz konuşmalar veya yazılar, konuşma veya metni dilimleme, anlam belirsizliği, söz dizimsel belirsizlik ve sözcük niteliklerindeki belirsizlik olarak sıralanabilir. Buna rağmen doğal dil işleme teknolojinin de gelişmesiyle birçok alanda kullanılmaktadır.

Doğal dil işleme için bazı kullanım şekilleri aşağıda açıklanarak verilmiştir (Adalı, 2012).

- Yazım yanlışlarının düzeltilmesi:

Günümüzde yazma eyleminin çoğunlukla bilgisayar ortamında yapılmasıyla beraber ihtiyaç duyulan bir kullanım şeklidir. İncelenecek metnin yazım dili bilgisayar veri tabanına geçirilerek uygulanmaktadır.

- Bul ve değiştir:

Metin içinde verilen herhangi bir kelimenin tamamının bulunup istenilen diğer bir kelime ile değiştirilmesidir. Burada değiştirilecek kelime ek aldıysa, değiştirilen kelimeye uygun ek getirilmesi ve bu yüzden ses bilimi özelliklerine göre düzeltme yapılması gerekmektedir.

- Basılı bir metni okuma:

Bilgisayar ortamında yer almayan basılı bir metnin optik olarak yazılı metinleri okuması ve okunan karakterlerin bilgisayar karakterine çevrilmesi işlemidir.

- Bir metnin özetini çıkarma:

İnternet ile bilgi sayısının artması, istenilen bilgiye doğru ve kısa bir şekilde ulaşmayı zorlaştırmaktadır. Bu sebeple metinlere dair özetleme işlemi yapılmaktadır. Bu işlem metne ait içindekiler/ başlık ve sık kullanılan kelimelerin incelenmesi ile gerçekleşmektedir.

- Metnin içerdiği bilgiyi çıkarma:

Doğal dillere ait kurallar kullanılarak yazılan metnin içinden içerdiği bilgiler çıkarılmaktadır. Tümceler incelenerek kişi, eylem, zaman ve yer gibi bilgiler aranmaktadır.

- Metni seslendirme:

Bilgisayara verilen yazılı metnin sesli olarak okuması işlemidir. Burada metne ait dilin ve ses biliminin bilinmesi gerekmektedir.

- Konuşmayı yazıya dökme:

Seslerin mikrofona ile elektriksel sinyale dönüştürülüp, sinyallerin metne dönüştürülmesi işlemidir. Türkçe gibi okunduğu gibi yazılan diller için bu işlem daha kolay gerçekleşmektedir.

- Soru yanıtlama:

Birçok sektörde hizmet vermek amacıyla kurulan çağrı merkezleri mevcuttur. Teknolojinin gelişmesiyle beraber artık bu işlemlerin bir kısmı da bilgisayarlar tarafından yapılmaktadır. Gerçekleştirilen uygulamalar ile müşterilerin konuşmalarından sözcükler çekilerek sözcüğe bağlı soru sorulan veya sözcüğe bağlı olarak yanıt verilen işlemidir. Bu işlem metni çevirme ve metni seslendirme işlemlerini içerir.

- Çeviri:

Diller arası çeviri için iki dile ait biçim, anlam ve ses bilimleri ve söz dizimi bilinmelidir. Ayrıca dillerin tam olarak bilinmesinin yanı sıra metnin konusu hakkında da bilgi sahibi olunmalıdır. Bilgisayar doğal dil işleme ile sözcük ve cümle çevirisini gerçekleştirebilmektedir.

- Metin işleme:

Metindeki sözcükleri kullanılacak amaç doğrultusunda kategorize etme, bölümlendirme ve kümeleme işlemidir.

Çalışma, doğal dil işlemenin uygulama alanlarından olan metin işleme ile kullanılarak gerçekleştirilmiştir. Bu yüzden bir sonraki adımda metin işleme biraz daha detaylı anlatılmıştır.

4.1.1. Metin işleme

Metin işleme, istenilen amaç doğrultusunda metne ait kelimeleri sınıflandırarak veya kümeleyerek analiz için hazır hale getirme işlemidir. Her çalışma veya amaç için kullanılacak yöntem farklı olmaktadır. Bu yüzden herhangi bir metne uygulanan kümeleme işlemi amaç değiştiğinde doğru sonuç vermeyebilir.

Metin işleme için kullanılan bazı teknikler aşağıda verilmiştir (http-3).

- Harf boyutunu düzenleme:

İncelenen metindeki tüm harfleri küçültme veya büyütme işlemidir. Bu sayede yapılacak analizlerde metnin nasıl yazıldığı durumu önemini yitirmektedir ve harf boyutları analizde aldatıcı bir duruma sebep olmamaktadır. Örneğin; “ANaliz” kelimesini “analiz” olarak düzenlenmektedir.

- Köke indirgeme:

İncelenecek metne ait tüm kelimeleri kök haline indirerek analiz sürecini kolaylaştıran ve doğruluk oranını arttıran bir işlemidir. Örneğin; “yapılmaktadır” kelimesi “yap” olarak değiştirmektedir.

- Bağlaç/edat kaldırma:

Metindeki bağlaç/ edatları kaldırıp analizi kolaylaştıran işlemidir. Örneğin; “doğal dil işleme nedir?” sorusu sadece “doğal dil işleme” olarak bırakılır.

- Normalizasyon:

Metindeki hatalı kelimeleri normalize işlemi ile olması gereken hale getirilmesidir. Örneğin; “haaarika” kelimesinin “harika” olarak, “3gen” kelimesinin “üçgen” olarak değiştirilmesi gibi.

- Gürültü kaldırma:

Metinde bulunan noktalama işaretleri ve sembollerin kaldırılması işlemidir.

- Metin zenginleştirme:

İncelenen metindeki kelimelerin yerine istenilen yeni kelimelerin yazılması işlemidir. Örneğin; “dil işleme” yerine “doğal dil işleme” yazılması.

Her zaman tüm metin işleme tiplerinin kullanılması gerekmemektedir. Ama etkili bir sonuç için bazılarının muhakkak yapılması gerekmektedir ve bunlar harf boyutunun düzenlenmesi ve gürültü kaldırma işlemleridir. Geriye kalan teknikler ise amaç doğrultusunda kullanılmalıdır.

4.2. Üretim Süreci

Üretim süreci, hammadde veya yarı mamullerin, üretim faktörleri aracılığıyla farklı bir çıktıya dönüştürme süreci, diğer bir ifade ile üretim aşamalarının tümü olarak tanımlanabilmektedir. Bu tanım ele alındığında üretim sürecinin 3 temel ögesi olduğu söylenebilir. Bunlar; girdiler, dönüşüm süreci ve çıktılarıdır.

Üretim süreçleri birçok farklı şekilde sınıflandırılmaktadır. Sınıflandırma yapılırken üretim yöntemleri, üretim miktarı, ürün türleri ve üretim akışı kullanılan ölçütlerdir. Üretim süreçlerinde yerleşim planlaması önemli bir yer tuttuğundan sınıflandırma ölçütü olarak yerleşim tipi çok fazla kullanılmaktadır.

Yerleşim tipine göre üretim süreçleri seri, atölye tipi, proje tipi, yalın ve hücre tipi üretim olarak sınıflandırılmaktadır. Çalışma kapsamında modellenen üretim süreçleri atölye tipi, seri ve hücre tipi üretimdir. Ayrıca yalın üretimde yerleşim tipine göre olan U- tipi üretim de modellenmiştir.

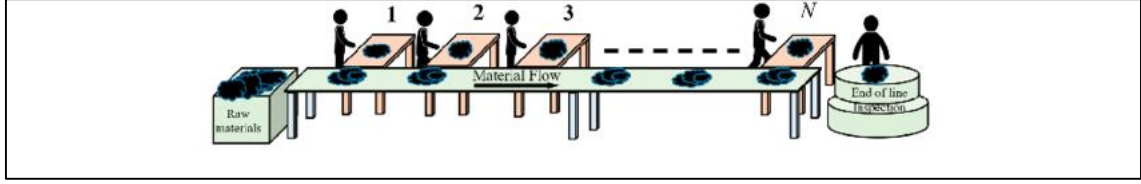
4.2.1. Seri üretim

Standart ve çok miktarda ürün üretilmesi için kullanılan seri üretim tipi, ardışık olarak sıralanmış işlem noktalarından oluşmaktadır. Bu üretim tipinde süreç standart ürün üretmek için tasarlandığından ürün üzerindeki değişikliklerde esnek bir yapı mevcut değildir. Fakat belirli işler için oluşturulan bu sistemlerin çalışma hızı ve kapasite kullanım oranları yüksektir.

Seri üretimde yerleşim düzeni ürünün işlem sırası dikkate alınarak yapılmaktadır. Hammaddeden ürüne kadar geçen sürede işlem görülecek makineler bir hat şeklinde ardışık olarak sıralanmaktadır. Bu durum seri üretim tipi için birçok avantaj sağlamaktadır. Üretim sırasında parçanın taşınacağı mesafe azdır. Ayrıca kapasite kullanım oranı fazla olduğundan kuyrukta bekleme süresi minimum düzeydedir. Bu durumlar da yarı mamul stoklarının azalmasını sağlamaktadır. Beklemelerin azalmasının bir diğer faydası ise toplam üretim süresinin kısalması ve birim zamanda üretilecek ürün sayısının artmasıdır.

Dezavantaj olarak ise ürüne göre yerleşim düzeninden dolayı ürün değişikliğinde hat üzerinde de değişiklikler gerekmektedir. Ayrıca üretimin akış hızının hattaki en yavaş makineye bağlı olması hat dengeleme problemini ortaya çıkarmaktadır. Herhangi bir makinede yaşanılacak bir arıza tüm hattın durmasına neden olabilmektedir. Ürün çeşidinin arttırılmak istenmesi sonucunda yeni bir üretim hattına ihtiyaç duyulmakta ve

bu durum yüksek yatırım maliyetine sebep olmaktadır. Şekil 4.1’ de bir seri üretim hattı örneği verilmiştir.



Şekil 4.1. N basamaklı seri üretim hattı

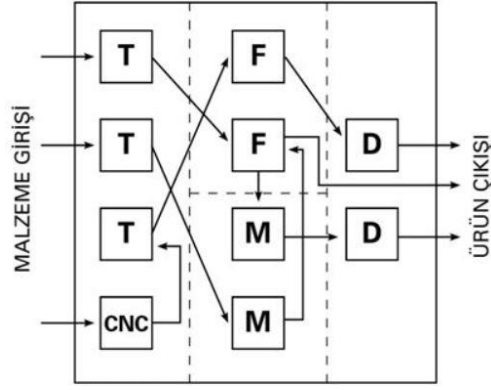
Verilen Şekil 4.1’de bir hat üzerinde N adet işlem basamağından geçen ürünün iş akışı gösterilmiştir. Bu üretim sisteminde ürün her işlem adımına uğramak zorundadır ve her işlem basamağından bir operatör sorumludur.

4.2.2. Atölye tipi üretim

Bu üretim tipinde genellikle çok amaçlı makine ve araç - gereçler bulunmakta ve bunlar özelliklerine göre gruplanmaktadır. Yani aynı makineler ile benzer işi yapabilen makineler aynı bölümde yer almaktadır. Sisteme gelen parçalar alternatif makinelerde işlenmekte eğer tüm makineler dolu ise parça kuyruğa alınmaktadır. Bu durum karmaşık bir iş akışına neden olmaktadır.

Atölye tipi üretimde üretilen ürün miktarı düşük olmakla beraber ürün çeşitliliği, iş istasyonlarında bekleme süresi ve ara stoklar fazladır. Bu da üretim süresini arttırmakta, birim zamanda üretilen ürün sayısını azaltmaktadır. Makinaların farklı işler için ayarlanması hazırlık süresini arttırmaktadır ve operatörün çok çeşitli parça işlemesinden dolayı verimi düşmektedir.

Şekil 4.2’de atölye tipi üretim sürecine ait bir örnek verilmiştir.



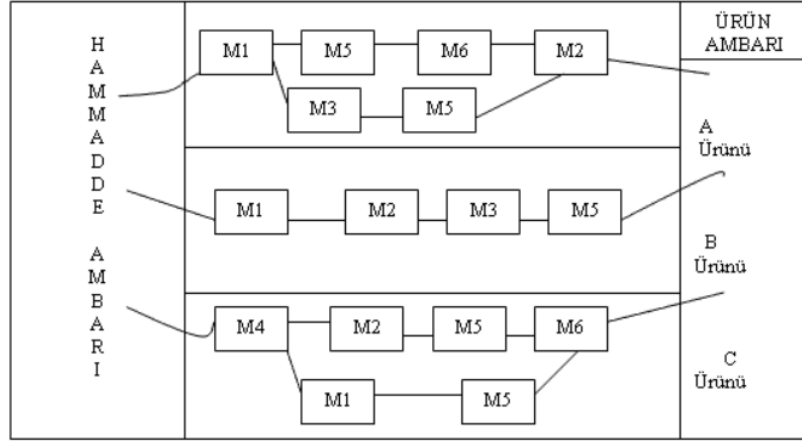
Şekil 4.2. Atölye tipi üretim süreci

Şekil 4.2’deki atölye tipi üretimde 5 tane farklı işlem merkezi verilmiş ve her işlem merkezine ait makineler bir arada gruplanmıştır. Her ürün tipine göre işlem basamakları değişkenlik gösterdiği için farklı iş akışları ortaya çıkmaktadır. Bu üretim tipinde ürün tüm işlem merkezlerine girmek zorunda değildir.

4.2.3. Hücre tipi üretim

Hücre tipi üretim, gruplardan oluşan ve her grupta benzer üretim karakterine sahip belirli parçaların üretimi için gerekli operatör, makine ve araç – gereçlerin olduğu üretim sistemidir. Her grup üretmesi gereken parçaları yeterli bir seviyede üretebilecek şekilde düzenlenmiştir.

Bu üretim sisteminde önce birbirleriyle benzer üretim işlemlerini gerektiren parça aileleri belirlenmektedir. Daha sonra belirlenen parça ailelerinin işlenmesi için gerekli olan makineler bir araya getirilerek hücreler oluşturulmaktadır. Şekil 4.3’te hücre tipi üretime ait görsel verilmiştir.



Şekil 4.3. Hücre tipi üretim süreci

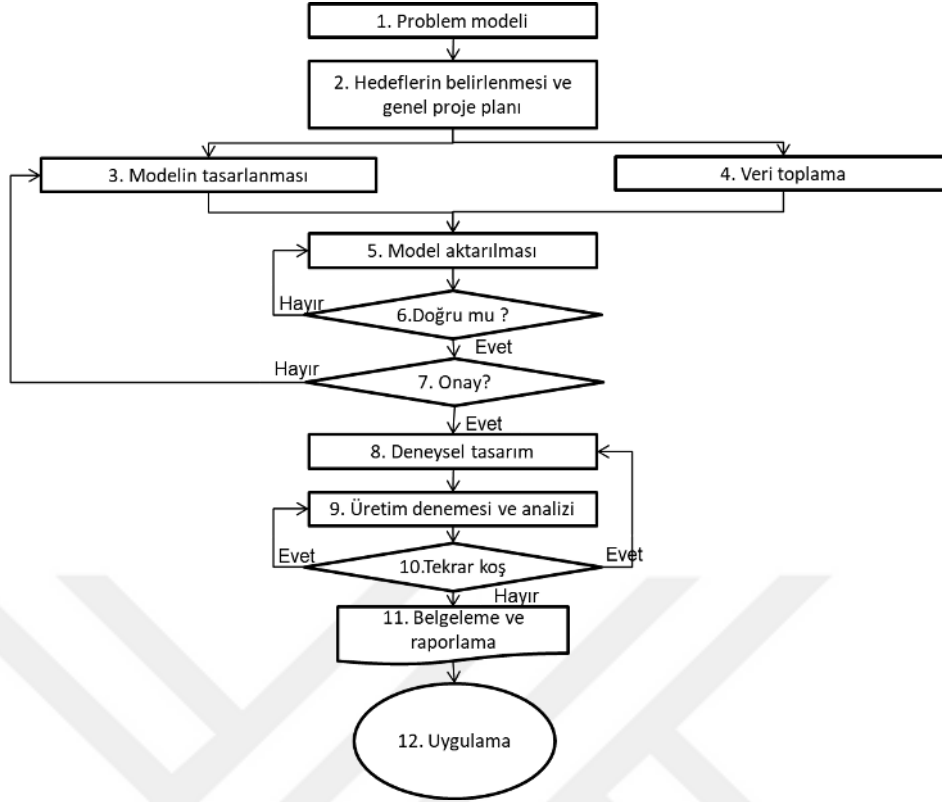
Şekil 4.3'te görüldüğü gibi her ürün tipini üretmek için farklı hücreler oluşturulmuştur. Bu hücrelerde üretilecek ürüne ait gerekli tüm makine çeşitleri bulunmaktadır. Bu sisteminde diğer üretim sistemlerine göre makine hazırlık ve toplam üretim süresi düşük, ara mamul stokları az ve parça akışı kolaydır. Ayrıca sorumlulukların alt sistemlere yüklenmesi operatör verimini arttırmaktadır.

4.2.4. U Tipi Hatlar

Japon işletmeler tarafından küçük parti üretimini sağlamak amacıyla geliştirilen Just in Time (JIT) ile sıfır stok seviyesine ulaşılmaya çalışılmaktadır. Temelde talebe göre üretim yapmayı hedefleyen bu yaklaşım ile talep edilen ürünlerin istenilen zaman ve miktarda, en iyi kalitede üretilmesi beklenmektedir.

JIT ile üretim içi stok düzeyi sıfır veya sıfıra yakın seviyede tutularak parçaların birbirini ardına işlendiği sürekli bir üretim sistemi oluşturulmaktadır. Bu amaçla pek çok yeni hat modelleri ileri sürülmüştür. Bu hat modellerinden biri olan U tipi hatlar ilk olarak Toyota'da JIT sisteminin uygulanmak istenmesiyle ortaya çıkmıştır.

Bu hat tipinde üretim işlemlerinin yapılış sırasına göre makineler U şekilli hattın etrafına dizilmektedir. Bu düzende bir işçi birden fazla makinadan sorumludur, tek parça akışı ve süreç bazlı hat anlayışı mevcuttur. Bir operatör ile hattın hem çıkışı hem de girişi kontrol edilmektedir. Hattın iki tarafında operatörlerin hareket edebilmesi ile toplam işçi sayısı azaltılmakta fakat bu durum operatörlerin çok fonksiyonlu olmalarını gerektirmektedir.



Şekil 4.5. Simülasyon adımlarına ait akış şeması

Gerçekleştirilmek istenen simülasyon için ilk önce problemin modeli tasarlanır. Ardından elde edilmek isten sonuçlar belirlenir. Daha sonra belirlenen problem ve amaçlarla simülasyon modeli tasarlanır ve veriler toplanır. Tasarlanan model bilgisayar ortamına aktararak modelde yazınsal bir hata olup olmadığına bakılır. Eğer hata varsa yazılan model üzerinde hatalar giderilmeye çalışılır eğer hata yoksa devam edilir. Bu aşamada ise tasarlanan modelin doğruluğu kontrol edilir. Eğer hata var ise modelin tasarlanma aşamasına geri dönülür, aksi halde ise model kabul edilip devam edilir. Sırasıyla model için deneysel tasarım ve simülasyonda üretim denemesi yapılır. Tekrar sayısı yeterli ise belgeleme ve raporlama aşamasına geçilir, yeterli değil ise deneysel tasarım veya üretim denemesi aşamasına dönülür. Simülasyon raporları incelendikten sonra amaçlar doğrultusunda gerçek model üzerinde uygulama aşamasına geçilir.

Gerçekleştirilen çalışmanın simülasyon aşamasına ait problem modeli, oluşturulan üretim süreçlerindeki akışı göstermektedir. Amaç ise simülasyon sonunda akışın verimini görmek ve bu verimi düşüren nedenleri tespit etmektir. Modelin tasarlanması ise kullanıcı tarafından girilen metin üzerinden oluşturulan süreç modelinden çekilir ve model, arka planda yazılan kodlar ile simülasyon aşamasına dönüştürülür. Veriye ait bilgiler ise

kullanıcı tarafından girilir. Eğer veri setlerine ait dağılım tipleri ve parametrelerine ait bilgiler bilinmiyor ise program aracılığıyla analiz edilebilir. Son olarak kullanıcı tarafından istenilen simülasyon süresi kadar model koşturulur ve amaç doğrultusundaki çıktılar raporlanır.

Çalışmada modellenen üretim süreçlerinin simülasyonunda makinelere ait işlem süreleri sabit bir değere sahip değildir yani belirsizdir ve bu durum gerçekleştirilen simülasyonu rassal simülasyon yapmaktadır. Bu yüzden değişkenleri tanımlayabilmek için olasılık dağılımlarına ihtiyaç duyulmaktadır. Bir sonraki başlıkta olasılık dağılımları incelenmiş ve kullanılan olasılık dağılımları açıklanmıştır.

4.3.1. Olasılık dağılımları

Olasılık dağılımı, bir rassal olayın ortaya çıkışını açıklayabilmek için gerekli değerleri ve olasılıkları tanımlamaktadır. Değerler olay için mümkün olan tüm sonuçları kapsamaktadır ve olasılıkların toplamı bire veya yüze eşit olmalıdır. Rassal değişken, bir rastgele olayın olası sonuçlarının nümerik değerleridir. Olasılık dağılımlarında kesikli ve sürekli olmak üzere iki çeşit rassal değişken bulunmaktadır.

Kesikli rassal değişkenler, sadece sayılabilir değerler alan yani alınabilecek tüm değerlerin sınırlı sayıda olan değişkenlerdir. Kesikli olasılık dağılımlarında en çok bilinen dağılımlar binom, poisson, bernouli ve geometrik olarak sıralanabilir.

Sürekli rassal değişkenler ise, sonsuz sayıda değerler alabilen değişkenleri ifade etmektedir. Sürekli dağılımlarda en çok kullanılanlar düzgün, üssel ve normal dağılımlardır. Çalışmada üretim süreleri rassal değişkenlerdir ve bu değişkenler için rassal değerler türetilmektedir. Bu rassal değerleri türetirken belirlenen aralıklarda sonsuz değerler alınabilen sürekli olasılık dağılımlarının en çok bilinen 3 çeşidi kullanılmıştır.

- Üssel dağılım:

Meydana gelen iki olay arasında geçen süre olarak tanımlanmaktadır. Olasılık yoğunluk fonksiyonu Denklem 1’de verilmiştir. Dağılımın parametresi (λ)’dır. $\lambda = (1/\mu)$ olarak tanımlanmakta ve sıfırdan büyük değerler almaktadır.

μ : ortalama

x : ilk durumun ortaya çıkması için gereken süre

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & , x > 0 \\ 0 & , d.d \end{cases} \quad (1)$$

- Uniform dağılım:

Eğer bir rassal değişken için olası değerlerinin ortaya çıkma ihtimalleri eşit ise bu rassal değişken uniform dağılıma sahiptir. Dağılıma ait olasılık yoğunluk fonksiyonu Denklem 2’de verilmiştir. Parametre değerleri a ve b’dir. Burada a, rassal değişkenin alabileceği en küçük değeri, b ise en büyük değeri ifade etmektedir.

$$f(x) = \begin{cases} \frac{1}{b-a} & , a < x < b \\ 0 & , d.d \end{cases} \quad (2)$$

- Normal dağılım:

Bu dağılım doğada çıkan olası sonuçları ifade etmektedir. Kesikli ve sürekli olasılık dağılımlarından birçoğu belirli şartlar altında normal dağılıma dönüşmektedir ve bu yüzden de sıklıkla normal dağılım kullanılmaktadır. Olasılık yoğunluk fonksiyonu Denklem 3’te verilmiştir. Dağılım parametreleri ortalama (μ) ve standart sapmadır (σ).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)}, \quad -\infty < x < \infty \quad (3)$$

Rassal simülasyon için kullanılacak verilerin dağılım tipinin bilinmesi gerekmektedir. Dağılım tipinin belirlenmesi için kullanılan testler vardır. Çalışmada veriye ait dağılım tipleri, en sık kullanılan Ki- kare ve Kolmogorov- Smirnov testleri kullanılarak belirlenmiştir. Bir sonraki adımda bu testler detaylı olarak incelenmiştir.

4.3.1.1. Ki- kare testi

Ki- kare testi iyi uyum, homojenlik ve bağımsızlık testleri olarak üçe ayrılmaktadır. Örnek verilerden yola çıkarak popülasyonun dağılımı hakkında ortaya atılan hipotezi test etmek amacıyla ki- kare uyum iyiliği testleri kullanılmaktadır. Önce H_0 ve H_1 hipotezleri oluşturulmaktadır.

H_0 : Veri ilgili dağılıma uymaktadır.

H_1 : Veri ilgili dağılıma uymamaktadır.

Frekanslar arasındaki uyum ki- kare testi ile ortaya konmaktadır. Bu durum test istatistiği olarak ifade edilmektedir. Sıfır hipotezi doğru ise gözlenen ve beklenen frekanslar arasında yakın bir ilişki olduğu söylenebilir. Test sonucu sıfıra yaklaştıkça uyum oranı artmaktadır. Denklem 4’te ki- kare test istatistiğine ait eşitlik verilmiştir.

O_i : Gözlenen değer

E_i : Beklenen değer

sd : Serbestlik derecesi ($r-1-m$)

r : Verideki kategori sayısı

m : Tahmin edilen parametre sayısı

$$\chi_{test}^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (4)$$

$$\chi_{test}^2 > \chi_{\alpha, sd}^2 \quad (5)$$

Denklem 4'te verildiği gibi hesaplanan ki- kare değeri tablodan elde edilen ki- kare değerinden büyük ise H_0 hipotezi reddedilir (Denklem 5). Yani bu durumda incelenen verinin ilgili dağılıma uymadığı söylenebilir.

4.3.1.2. Kolmogorov- Smirnov testi

Kolmogorov- Smirnov testi ki- kare testinin bir alternatifidir. Ki- kare testinde daha doğru sonuç için beklenen frekans değerlerinin 5'ten büyük olması gerekmekte eğer 5'ten küçük frekans değerleri mevcut ise örnek sayısının çok fazla olması istenmektedir. Fakat bu durum Kolmogorov- Smirnov uyum testi için söz konusu olmaması bu testi daha kullanışlı kılmaktadır (Kartal, 1998).

Burada da ki- kare testinde olduğu gibi iki hipotez oluşturulur.

H_0 : Veri ilgili dağılıma uymaktadır.

H_0 : Veri ilgili dağılıma uymamaktadır.

Test istatistiği (D), gözlenen ve beklenen değerlerin kümülatif nisbi frekansları arasındaki farkın mutlak değerce en büyüğüne eşittir (Kartal, 1998). Test istatistiğine ait eşitlik Denklem 6'da verilmiştir.

$$D = \max|F_O - F_E| \quad (6)$$

F_O : Gözlenen kümülatif nisbi frekans

F_E : Beklenen kümülatif nisbi frekans

Denklem 6'daki değer hesaplandıktan sonra tablodan $D_{\alpha, n}$ değeri elde edilir. Bu tablo değeri hesaplanan test istatistiğinden (D) küçük ise H_0 hipotezi reddedilir. Yani gözlenen frekansların beklenen frekans değerlerine uygun olmadığı sonucuna varılır.

4.4. Kullanılan Kütüphaneler

Gerçekleştirilen çalışma Python 3.7. kullanılarak kodlanmış ve programa ait birçok kütüphaneden yararlanılmıştır. Problemin çözümü için kullanılan kütüphaneler ve ne amaçla kullanıldığı aşağıda verilmiştir.

- wxPython:

Python programlama dili için platformlar arası bir grafik kullanıcı ara yüzü kütüphanesidir. Python gibi wxPython da açık kaynak kodlu bir yazılımdır ([http-4](http://4)). Herkes tarafından kullanılabilir, paylaşılabılır ve düzenlenebilir. Çalışmada kullanıcı ile etkileşimi sağlamak için gerçekleştirilen ara yüzlerin tamamı bu kütüphane kullanılarak oluşturulmuştur.

- Natural Language Toolkit (NLTK) :

Bird vd. (2009) tarafından yayınlanan kitapta Python için NLTK kütüphanesi anlatılmıştır. Bu kütüphane ile metin işleme teknikleri olan sınıflandırma, etiketleme, kaynak oluşturma, ayrıştırma ve anlamsal akıl yürütme gerçekleştirilebilir. Çalışmada girdi olan metni düzenleyerek cümle ve kelimelere ayırmak, karşılaştırma ve sınıflandırma yapmak için kullanılmıştır.

- TurkishStemmer:

Türkçe kelimeleri köklerine ayırmak için kullanılan kütüphanedir ([http-5](http://5)). Çalışmada girdi olan metin nltk kütüphanesi ile kelimelere ayrılmış, daha sonra TurkishStemmer kullanılarak bu kelimeler köklere indirgenmiştir.

- Numpy:

Oliphant (2006) ve van der Walt vd. (2011) tarafından geliştirilen Numpy kütüphanesi anlatılmıştır. Bilimsel hesaplamalarda n boyutlu diziler oluşturmak ve dizi işlemleri gerçekleştirmek için kullanılmaktadır. Çalışmada simülasyon aşamasında dağılım tipine göre rastgele sayı üretme ve dizi işlemlerinde kullanılmıştır.

- SciPy:

Bilimsel hesaplamalarda ve mühendislik hesaplamalarında kullanılan açık kaynak kodlu bir kütüphanedir ([http-6](http://6)). Çalışmanın simülasyon ve veri analizi aşamasında kullanılan kütüphane ile Ki- kare testi ve Kolmogorov- Smirnov testi gerçekleştirilmiştir.

- Matplotlib:

Hunter (2007) tarafından yayınlanan makale ile tanıtılan Matplotlib kütüphanesi, Python programlama dili için iki boyutlu ve yüksek çözünürlüklü veri görselleştirme

aracıdır. Açık kaynak kodlu bir kütüphanedir. Çalışmada Matplotlib, simülasyon çıktılarının ve veri analizindeki girdilerin görselleştirilmesinde kullanılmıştır.

- Simpy:

Süreç tabanlı ayrık olay simülasyonları için kullanılan açık kaynak kodlu bir kütüphanedir ([http-7](http://7)). Çalışmada simülasyon aşamasında süreçlerin tanımlanmasında, hammadde ve insan kaynağının süreçle olan etkileşimin oluşturulmasında kullanılmıştır.



5. UYGULAMA

Python 3.7. ile kodlanan ve Bölüm 4.4'te tanımlanan kütüphaneler kullanılarak oluşturulan programın çalışma şekli aşağıdaki adımlarda sıralı olarak verilmiştir. Ayrıca programın çalışma şekline dair akış şeması Şekil 5.1'de görselleştirilmiştir.

Adım 1: Girdi olarak kullanılacak metin, kullanıcı tarafından ana sayfa ara yüzündeki gerekli yere yazılır veya metin dosyası seçilir.

Adım 2: Girilen metin program tarafından öncelikle tüm harf boyutu küçültülecek şekilde düzenlenir.

Adım 2: Metin kelimelerine ayrılır ve noktalama işaretlerinden temizlenir.

Adım 3: Her kelimenin kökleri belirlenerek yeni bir veri seti oluşturulur.

Adım 4: Oluşturulan model ile veri setinin analizi yapılır ve üretim sürecinin 4 farklı senaryodan hangisine uyduğu belirlenir.

Adım 5: Programda yer alan makine havuzu ile veri seti karşılaştırılarak makineler belirlenip sınıflandırılır.

Adım 6: Sınıflandırılan makineler Adım 4'te belirlenen senaryoya göre yerleştirilir ve üretim süreci modellenir.

Adım 7: Oluşturulan model kaydedilir/ yazdırılır.

Adım 8: Adım 15'e geçilir veya simülasyon aşaması ile devam edilir.

Adım 9: Veri analizi gerçekleştirilir veya Adım 11'e geçilir.

Adım 10: Veri seti seçilir ve verinin 3 dağılım tipinden hangisine uyduğu belirlenir.

Adım 11: Makine işlem sürelerine ait dağılım tipleri ve parametreleri kullanıcı tarafından simülasyon ara yüzünde gerekli yerlere girilir.

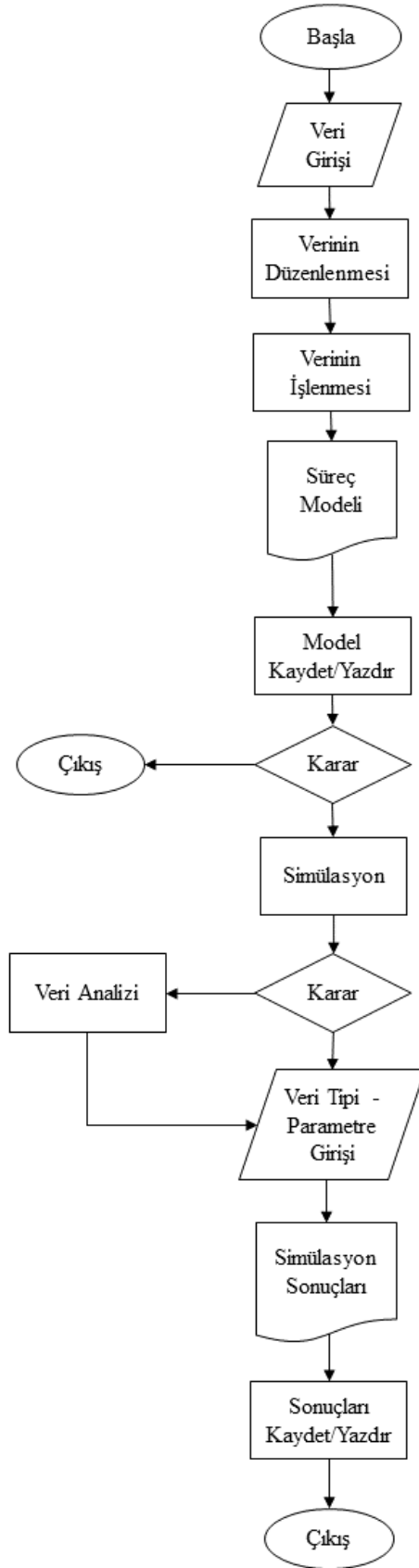
Adım 12: Simülasyon gerçekleştirilir.

Adım 13: Simülasyon sonuçları grafik halinde oluşturulur.

Adım 14: Sonuçlar kaydedilir/yazdırılır.

Adım 15: Programdan çıkılır.

Yukarıdaki adımlarla programın işleyişi en basit haliyle anlatılmıştır. Bu bölüme ait alt başlıklarda ise süreç modelleme, simülasyon ve veri analizi daha detaylı açıklanmıştır.



Şekil 5.1. Programa ait akış şeması

5.1. Süreç Modelleme

Herhangi bir ürünün yapılışı veya üretilişi bir süreç oluşturmaktadır. Bu çalışmada fabrika ortamındaki üretim süreci ele alınmıştır. Süreç modellenmesinin işleyişini anlatmak için Python’da oluşturulan algoritmaya ait sözde kod parçalı bir şekilde ve açıklanarak aşağıda verilmiştir.

```
Adım 1: Süreci anlatan metni gir.  
Adım 2: Metni çek ve text=[] matrisinin içine at.  
Adım 3: text matrisindeki metni kelimelere ayır ve her kelimeyi word_list=[] matrisinin içine at.  
Adım 4: word_list matrisindeki her kelimeyi küçük harfe dönüştür.  
Adım 5: word_list matrisindeki her kelimenin eklerini at ve kök halindeki kelimeleri s_list=[] matrisine at.
```

Şekil 5.2. Süreç modelleme algoritmasına ait ilk 5 adım

Şekil 5.2’de verilen sözde kod ile girilen metin çekilir ve işlenmek üzere bir matrisin içine atılır. Daha sonra metin kelimelere ayrılır ve ayrılan her kelime üzerinde harf düzenlemesi yapılır. Harfleri küçültülen kelimelerin ekleri atılarak her kelime kök haline indirgenir.

```
Adım 6: tip_list(3x3) ve s_list matrislerindeki kelimelere bak;  
    Eğer eşleşen kelime matrisin birinci satırındaysa;  
        Eşleşen kelimedenden önceki kelimeye bak ve kelime ‘u’ ise u_list=[]  
        matrisine kelimeyi ekle.  
    Eğer eşleşen kelime matrisin ikinci satırındaysa;  
        Eşleşen kelimedenden önceki kelimeye bak ve kelime ‘hücre’ ise hucre_list=[]  
        matrisine kelimeyi ekle.  
    Eğer eşleşen kelime matrisin üçüncü satırındaysa;  
        Eşleşen kelimedenden sonraki kelimeye bak ve kelime ‘hat’ ise seri_list=[]  
        matrisine kelimeyi ekle.  
    Değilse;  
        Adım 7’ye geç.
```

Şekil 5.3. Süreç modelleme algoritmasına ait 6. adım

Şekil 5.3’te verilen 6. adımda ise verilen metindeki sürecin hangi üretim tipine ait olduğu belirlenmeye çalışılır. Seri üretim için metinde hat kelimesinin varlığı aranır. Eğer mevcut ise ondan önceki kelimenin ilk, son veya herhangi bir seviye ifadesi (1., ikinci

vb.) olup olmadığına bakılır. Bu durumlardan biri varsa üretimin seri üretim olduğu belirlenir ve bir sonraki adımla devam edilir. Aksi durumda ise bir sonraki senaryo olan hücre tipinin varlığı araştırılır. Metinde ilk önce hücre kelimesi aranır. Eğer mevcut ise bir sonraki kelimenin imalat, üretim veya tip olup olmadığına bakılır. Bu durumlardan birinde senaryo kabul edilir aksi durumda bir sonraki senaryo olan U tipine bakılır. Burada ise “u” harfinin araması yapılır. Eğer işlenen veride tek başına bir “u” harfi bulunur ise bir sonraki kelime olarak tip ve hat kelimeleri aranır ve durum mevcut ise süreç U tipi üretim olarak tanımlanır. Tüm bu durumların mevcut olmaması halinde ise üretimin atölye tipi üretim olduğu kabul edilerek akış devam ettirilir.

Üretim sürecine ait senaryo belirlendikten sonra metin içerisindeki makineler aranır. Bu işlem programda mevcut olan makine havuzundaki kelimelerle metindeki kelimelerin karşılaştırılmasıyla gerçekleştirilir. Eğer havuzda olmayan bir makinenin metinde yer aldığı düşünülüyor ise bu makine kullanıcı tarafından sisteme eklenebilir. Eşleştirilen makineler kullanılan üretim süreci tipine göre farklı şekilde sınıflandırılır. Örneğin, her hattaki makinelerin ayrı ayrı sınıflandırılması gibi. Daha sonra sınıflandırılan makineler süreç şekline göre yerleştirilir. Belirlenen sürece göre makineler yerleştirildikten sonra başka bir üretim adımı olup olmadığı aranır, durum mevcut ise makineler sürece işlenir. Örneğin; seri üretim tipinde hatlarda üretilen ürünlerin bir sonraki adımda montaja gitmesi gibi.

Şekil 5.4’te verilen algoritmanın 7. adımında ise üretim tipinin U tipi olduğu durum ele alınmıştır. Tipi belirlenen süreç için daha sonra makine tespiti yapılır ve makine sayısına göre U şeklinde bir hat çizilir. Bu çizim eğer makine sayısı çift ise kısa kenarda iki aksi halde kısa kenarda bir makine olacak şekilde çizilir ve metindeki sıraya göre makineler yerlerine yerleştirilir. Daha sonra hattın bitiminde bir işlem olup olmadığına bakılır.

Adım 7: Eğer $u_list > 0$ ise;

Hattan sonra sürecin devam edip etmediğini kontrol etmek için hat_list1 ve s_list matrislerindeki kelimelere bak.

Eğer eşleşen kelime var ise;

Eşleşen kelimedenden sonra gelen kelime 'son' ise kelime konumunu $h_list2 = []$ matrisine ekle.

Değilse;

Devam et.

Eğer $h_list2 > 0$ ise;

h_list2 matrisinde bulunan kelime konumuna kadar s_list matrisindeki kelimeler ile mac_list matrisindeki kelimeleri karşılaştır ve eşleşen kelimeleri $surec = []$ matrisine ekle. h_list2 matrisinde bulunan kelime konumundan itibaren s_list matrisindeki kelimeler ile mac_list matrisindeki kelimeleri karşılaştır ve eşleşen kelimeleri $surec1 = []$ matrisine ekle.

Değilse;

s_list matrisindeki kelimeler ile mac_list matrisindeki kelimeleri karşılaştır ve eşleşen kelimeleri $surec = []$ matrisine ekle.

Eğer $surec$ matrisindeki kelime sayısı çift ise;

Kısa kenara iki makine gelecek şekilde bir U hattı çiz ve sırayla yerleştir.

Değilse;

Kısa kenara bir makine gelecek şekilde bir U hattı çiz ve sırayla yerleştir.

Eğer $surec1 > 0$ ise;

$surec1$ matrisinde yer alan her kelime için hattın altında bir işlem noktası çiz ve kelimeleri yaz.

Değilse;

Adım 8'e geç.

Şekil 5.4. Süreç modelleme algoritmasına ait 7. adım

Adım 8: Eğer seri_list>0;

hat_list ve s_list matrisindeki kelimeleri karşılaştır ve eşleşen kelimedenden sonraki kelimeye bak, kelime hat ise eşleşen kelime konumunu h_list1=[] matrisine ekle.

Hattan sonra sürecin devam edip etmediğini kontrol etmek için hat_list1 ve s_list matrislerindeki kelimelere bak.

Eğer eşleşen kelime var ise;

Eşleşen kelimedenden sonra gelen kelime 'son' ise kelime konumunu h_list2=[] matrisine ekle.

Değilse;

Devam et.

Eğer h_list2>0 ise;

hat_list2 matrisinde bulunan kelime konumuna kadar, s_list matrisindeki kelimeleri h_list1 matrisinde verilen her bir konum aralığı için mac_list matrisindeki kelimeler ile karşılaştır ve eşleşen kelimeleri konumuna göre her biri ayrı satırda olacak şekilde surec=[] matrisine ekle. h_list2 matrisinde bulunan kelime konumundan itibaren s_list matrisindeki kelimeleri mac_list matrisindeki kelimeler ile karşılaştır ve eşleşen kelimeleri surec1=[] matrisine ekle.

Değilse;

s_list matrisindeki tüm kelimeleri h_list1 matrisinde verilen her bir konum aralığı için mac_list matrisi ile karşılaştır ve eşleşen kelimeleri konumuna göre her biri ayrı satırda olacak şekilde surec=[] matrisine ekle.

surec matrisinde bulunan satır sayısı kadar hat oluştur.

Her hatta satırdaki eleman kadar makine noktası çiz.

Her makine noktasına sırayla isimlerini yaz ve aralarına akışı gösteren ok çiz.

Eğer surec1>0 ise;

surec1 matrisinde yer alan her kelime için hatların sağında alt alta gelecek şekilde işlem noktalarını çiz ve isimlerini yaz.

Değilse;

Adım 9'a geç.

Şekil 5.5. Süreç modelleme algoritmasına ait 8. adım

Şekil 5.5'te verilen sözde kodun 8. adımında ise süreç tipinin seri üretim tipi olduğu durumlar ele alınmıştır. Burada ilk önce hat sayısı daha sonra ise makineler belirlenir ve makineler bulundukları hat konumuna göre sınıflandırılır. Daha sonra hat ve makineler çizilerek her makine sırasıyla yerine yerleştirilir. Hatların sonunda parçaların işlem

göreceği başka noktaların olup olmadığı aranır ve mevcut ise işlem noktaları da hatların en sağından itibaren alt alta yerleştirilir.

```
Adım 9: Eğer hücre_list>0 ise;
    hücre_list ve s_list matrisindeki kelimeleri karşılaştır ve eşleşen
    kelimedenden sonraki kelimeye bak, kelime hücre ise eşleşen kelime konumunu
    h_list1=[] matrisine ekle.
    Hücrelerden sonra sürecin devam edip etmediğini kontrol etmek için
    hücre_list1 ve s_list matrislerindeki kelimelere bak.
    Eğer eşleşen kelime var ise;
        Eşleşen kelimedenden sonra gelen kelime 'son' ise kelime konumunu
        h_list2=[] matrisine ekle.
    Değilse;
        Devam et.
    Eğer h_list2>0 ise;
        hat_list2 matrisinde bulunan kelime konumuna kadar, s_list
        matrisindeki kelimeleri h_list1 matrisinde verilen her bir konum
        aralığı için mac_list matrisindeki kelimeler ile karşılaştır ve
        eşleşen kelimeleri konumuna göre her biri ayrı satırda olacak
        şekilde surec=[] matrisine ekle. h_list2 matrisinde bulunan kelime
        konumundan itibaren s_list matrisindeki kelimeleri mac_list
        matrisindeki kelimeler ile karşılaştır ve eşleşen kelimeleri
        surec1=[] matrisine ekle.
    Değilse;
        s_list matrisindeki tüm kelimeleri h_list1 matrisinde verilen her
        bir konum aralığı için mac_list matrisi ile karşılaştır ve eşleşen
        kelimeleri konumuna göre her biri ayrı satırda olacak şekilde
        surec=[] matrisine ekle.
    adet_list ile s_list matrisini karşılaştır.
```

Şekil 5.6. Süreç modelleme algoritmasına ait 9. adım

Üretim tipinin hücreyel imalat olması durumunda Şekil 5.6 ve Şekil 5.7’de verilen sözde kod kullanılmaktadır. Sırasıyla hücre sayısının belirlenmesi, her hücredeki makine tiplerinin belirlenmesi ve hücre sonlarında herhangi bir işlem noktasının varlığının olup olmadığına dair işlemler gerçekleştirilir. Daha sonra hücrelerdeki her makine tipinden kaç adet olduğuna bakılır ve belirlenen miktarlar matrislere işlenir. Sayısı kadar hücre ve her hücreye adedi kadar makine çizilir. Son olarak başka bir işlem noktası var ise hücrelerin en sağına alt alta sırasıyla çizilir.

```

Eğer eşleşen kelime var ise;
    Eğer sonraki kelime adet veya tane ise;
        Bir sonraki kelimeyi surec2=[] matrisine ekle.
        adet_list matrisinde eşleşen satırdaki ilk elemanı
        adet_list1=[] matrisine ekle.
    Eğer sonraki kelime surec matrisindeki herhangi bir kelime ise;
        Kelimeyi surec2=[] matrisine ekle.
        adet_list matrisinde eşleşen satırdaki ilk elemanı
        adet_list1=[] matrisine ekle.
    Değilse;
        Devam et.
surec matrisinde bulunan satır sayısı kadar hücre oluştur.
surec2 matrisindeki kelime konumu ile adet_list1 matrisinde aynı
konumdaki elemanı çek.
Yazan sayının bir eksiği kadar surec2 matrisindeki kelimedenden surec
matrisine ekle.
Her hücreye satırdaki eleman sayısı kadar makine noktası çiz.
Her makine noktasına sırayla isimlerini yaz.
Eğer surec1>0 ise;
    surec1 matrisinde yer alan her kelime için hücrelerin sağında alt
    alta gelecek şekilde işlem noktalarını çiz ve isimlerini yaz.
Değilse;
    Adım 10'a geç.

```

Şekil 5.7. Süreç modelleme algoritmasına ait 9. adımın devamı

Son adımın verildiği Şekil 5.8’de ise atölye tipine göre süreç modellenir. Burada ilk önce makine tipleri belirlenip bir sınıfta toplanır. Daha sonra ise her makine tipinden kaç adet olduğu tespit edilir ve her makine tipinin adedi sırasıyla bir matrise işlenir. Çizim aşamasında ise her makine tipi için bloklar oluşturulur ve her blokta makine tipinden adedine göre çizilir. Son olarak makine isimleri yazılır.

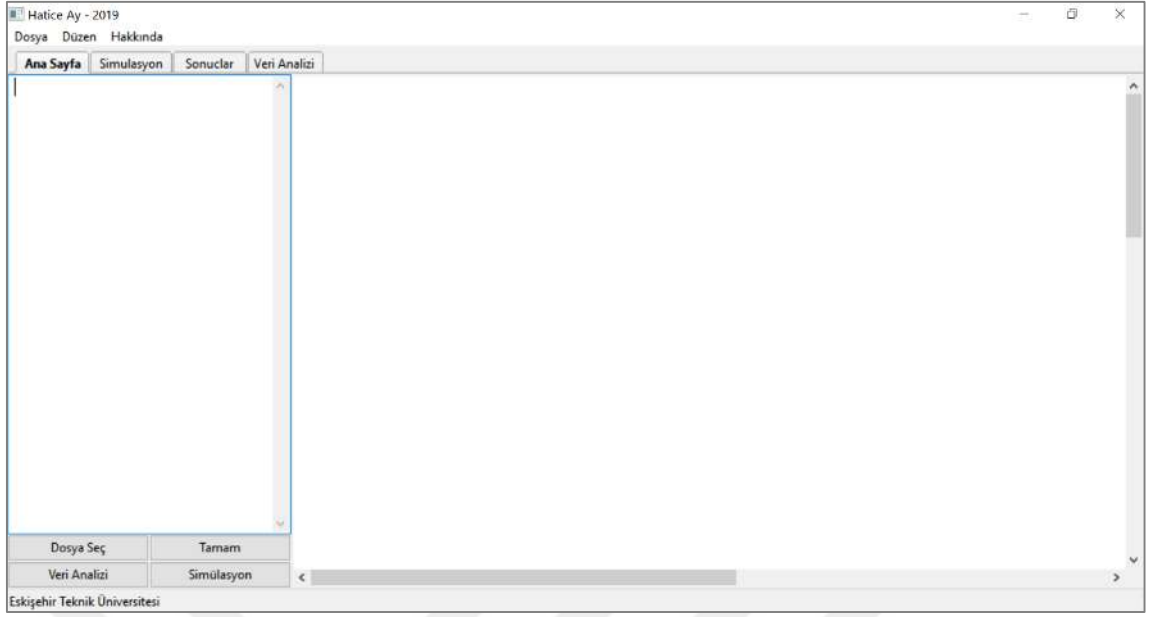
```

Adım 10: Değilse;
    s_list matrisindeki kelimeler ile mac_list matrisindeki kelimeleri
    karşılaştır ve eşleşen kelimeleri surec=[] matrisine ekle.
    adet_list ile s_list matrisini karşılaştır.
    Eğer eşleşen kelime var ise;
        Eğer sonraki kelime adet veya tane ise;
            Bir sonraki kelimeyi surec2=[] matrisine ekle.
            adet_list matrisinde eşleşen satırdaki ilk elemanı
            adet_list1=[] matrisine ekle.
            Eğer sonraki kelime surec matrisindeki herhangi bir kelime
            ise;
                Kelimeyi surec2=[] matrisine ekle.
                adet_list matrisinde eşleşen satırdaki ilk elemanı
                adet_list1=[] matrisine ekle.
        Değilse;
            Devam et.
    surec2 matrisindeki eleman konumu ile adet_list1 matrisinde aynı
    konumdaki elemanı çek.
    Yazan sayının bir eksiği kadar surec2 matrisindeki kelimedenden surec
    matrisine ekle.
    surec matrisinde yer alan aynı kelimeleri gruplayarak çiz.

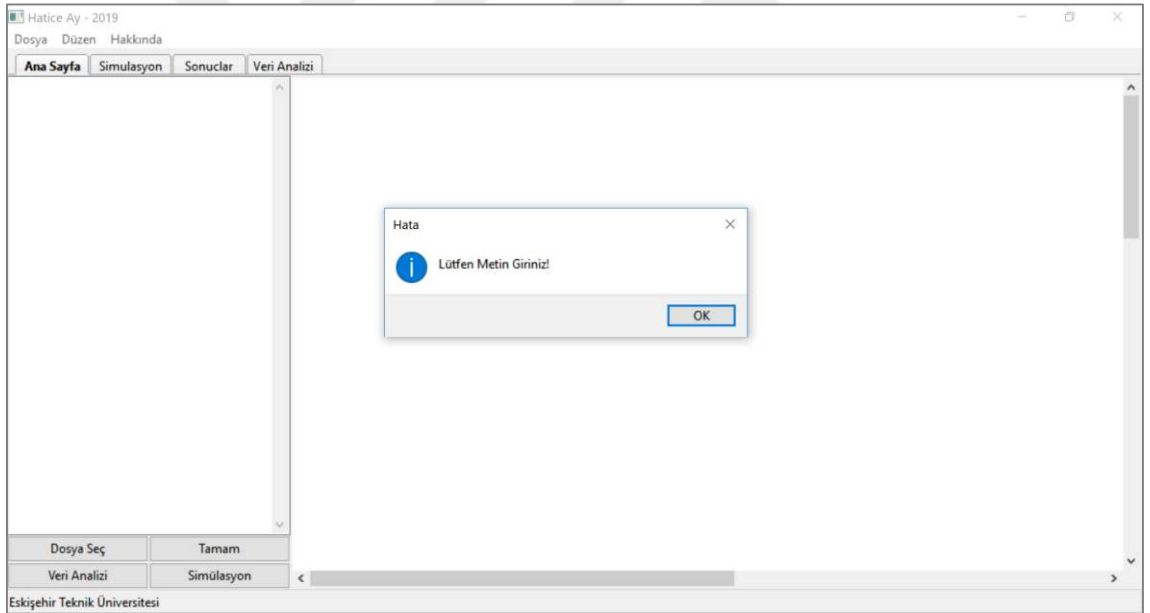
```

Şekil 5.8. Süreç modelleme algoritmasına ait 10. adım

Program için yazılan kod en basit haliyle sözde kod ile anlatılmıştır. Fakat kodun aktif hale gelebilmesi için programın “Ana sayfa” ara yüzü kullanılmalıdır. Programın “Ana sayfa” ara yüzü Şekil 5.9’da verilmiştir. Solda bulunan yazım alanına metin girilir veya sol altta bulunan “Dosya Seç” butonu ile metin dosyası seçilebilir ve seçildikten sonra metin yazım alanında görüntülenir. Daha sonra “Tamam” butonuna basılarak veri işlenmeye başlanır. Eğer herhangi bir metin girilmeden butona basılırsa Şekil 5.10’daki uyarı ile karşılaşılır.

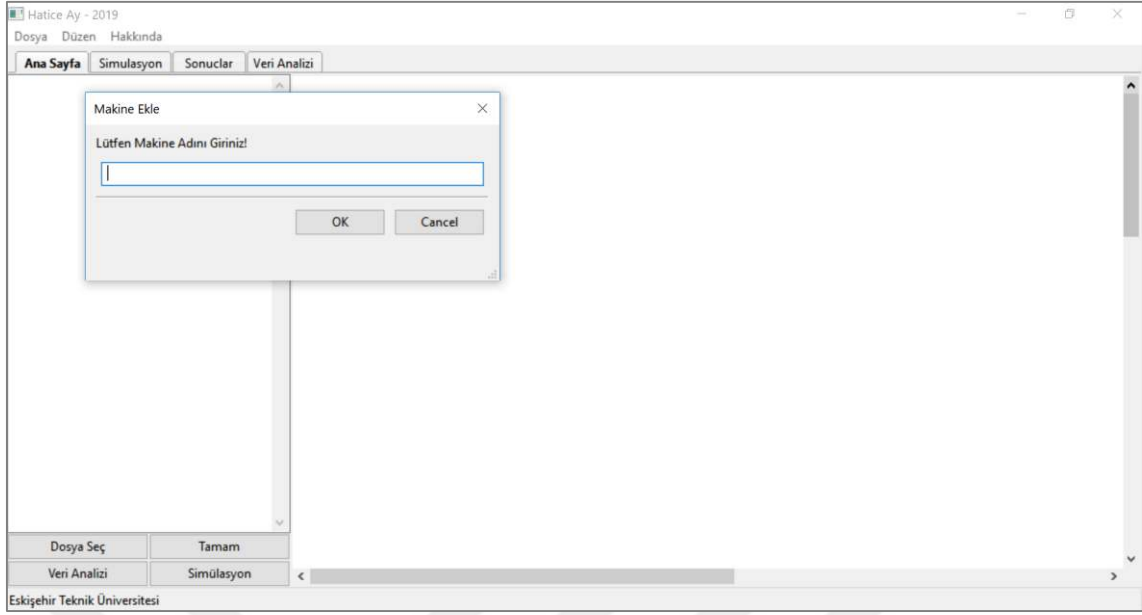


Şekil 5.9. Ana sayfa ara yüzü

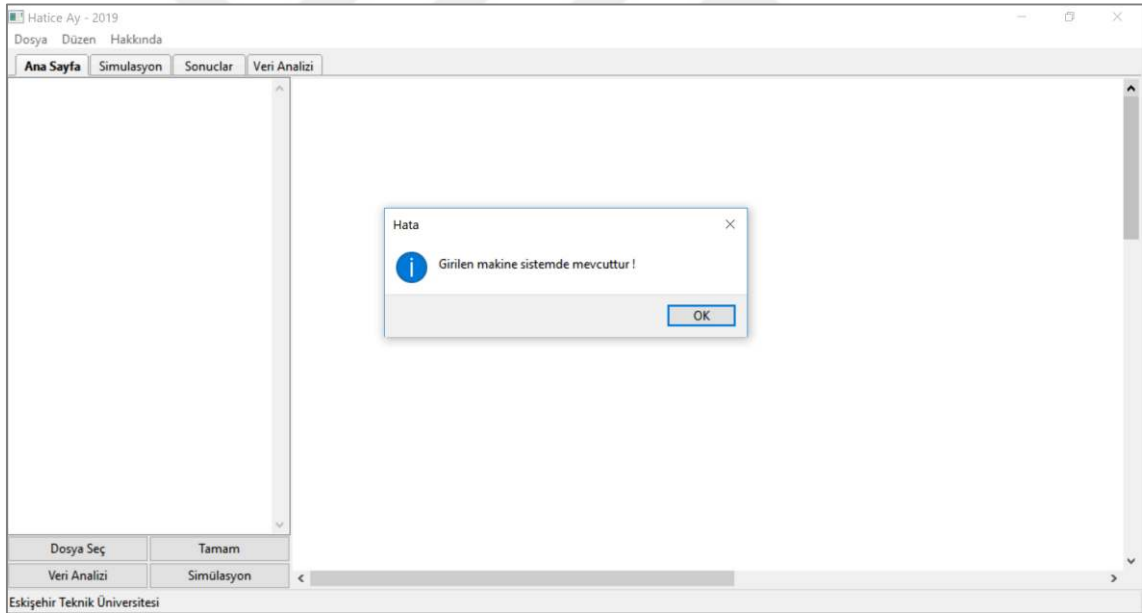


Şekil 5.10. Metin giriniz uyarısı

Sisteme makine eklemek isteniyorsa bunun için sol üstte bulunan “Düzen” sekmesinin içinde yer alan “Makine Ekle” butonu kullanılır. Daha sonra Şekil 5.11’de görünen metin giriş alanı ile karşılaşılır. Girilen makine sistemde mevcut değil ise makine havuzuna eklenir aksi halde Şekil 5.12’deki uyarı ekranı ile karşılaşılır.



Şekil 5.11. Makine ekleme alanı



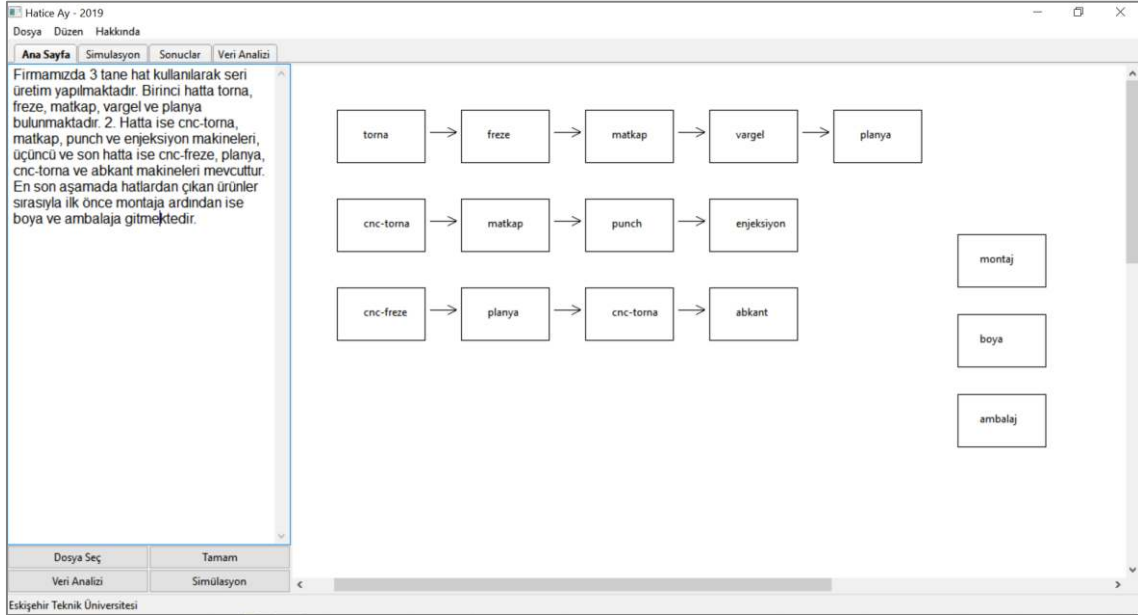
Şekil 5.12. Makine ekleme uyarısı

Her senaryo tipi için örnek metinler ve bu metinler işlenerek elde edilen süreç modellerine ait görseller elde edilmiştir.

Metin 1: Firmamızda 3 tane hat kullanılarak seri üretim yapılmaktadır. Birinci hatta torna, freze, matkap, vargel ve planya bulunmaktadır. 2. Hattı ise cnc-torna, matkap, punch ve enjeksiyon makineleri, üçüncü ve son hatta ise cnc-freze, planya, cnc-torna ve

abkant makineleri mevcuttur. En sonda hatlardan çıkan ürünler sırasıyla ilk önce montaja ardından ise boya ve ambalaja gitmektedir.

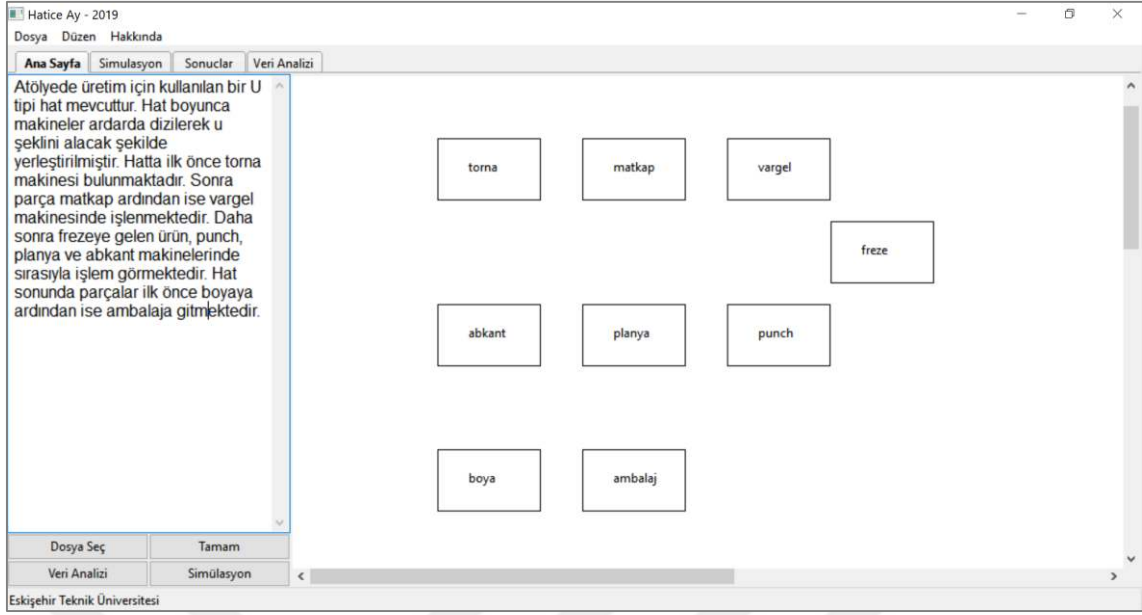
Metin 1'e ilişkin elde edilen görsel Şekil 5.13'te verilmiştir.



Şekil 5. 13. Metin 1 ile elde edilen model

Metin 2: Atölyede üretim için kullanılan bir U tipi hat mevcuttur. Hat boyunca makineler artarda dizilerek u şeklini alacak şekilde yerleştirilmiştir. Hatta ilk önce torna makinesi bulunmaktadır. Sonra parça matkap ardından ise vargel makinesinde işlenmektedir. Daha sonra frezeye gelen ürün, punch, planya ve abkant makinelerinde sırasıyla işlem görmektedir. Hat sonunda parçalar ilk önce boyaya ardından ise ambalaja gitmektedir.

Metin 2'ye ilişkin elde edilen görsel Şekil 5.14'te verilmiştir.



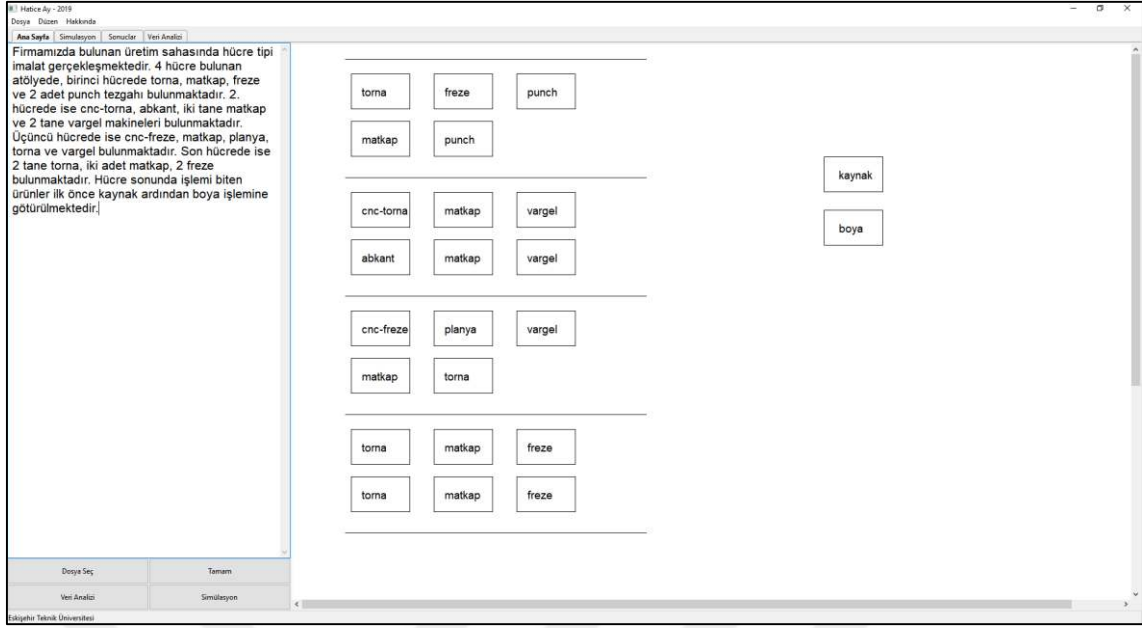
Şekil 5.14. Metin 2 ile elde edilen model

Metin 3: Firmamızda bulunan üretim sahasında hücre tipi imalat gerçekleştirilmektedir. 4 hücre bulunan atölyede, birinci hücrede torna, matkap, freze ve 2 adet punch tezgahı bulunmaktadır. 2. hücrede ise cnc-torna, abkant, iki tane matkap ve 2 tane vargel makineleri bulunmaktadır. Üçüncü hücrede ise cnc-freze, matkap, planya, torna ve vargel bulunmaktadır. Son hücrede ise 2 tane torna, iki adet matkap, 2 freze bulunmaktadır. Hücre sonunda işlemi biten ürünler ilk önce kaynak ardından boya işlemine götürülmektedir.

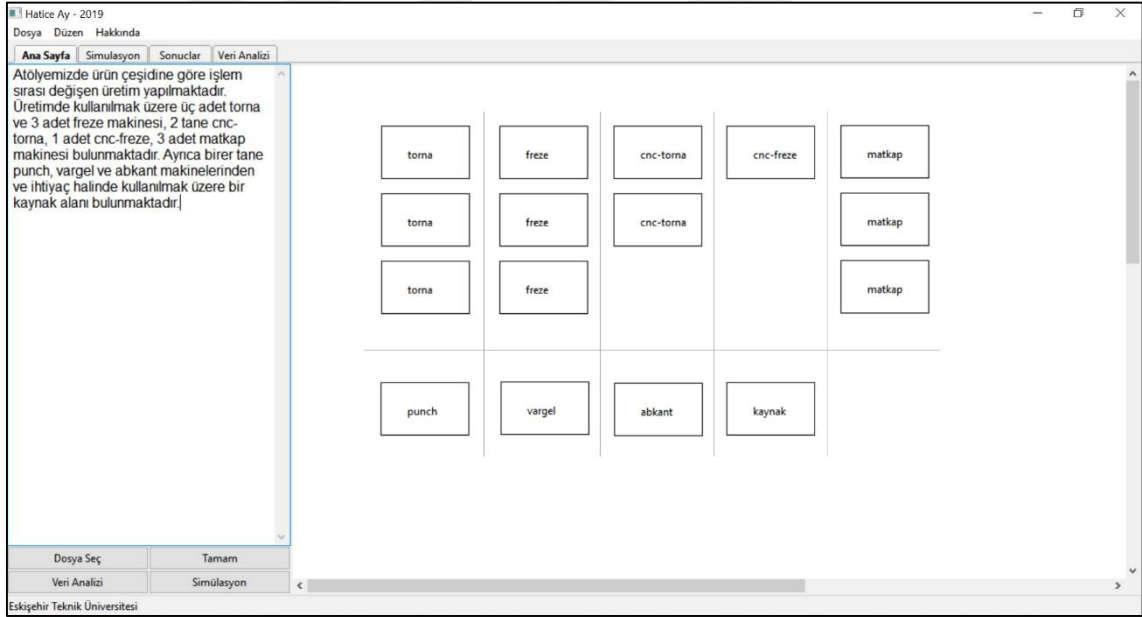
Metin 3'e ilişkin elde edilen görsel Şekil 5.15'de verilmiştir.

Metin 4: Atölyemizde ürün çeşidine göre işlem sırası değişen üretim yapılmaktadır. Üretimde kullanılmak üzere üç adet torna ve 3 adet freze makinesi, 2 tane cnc-torna, 1 adet cnc-freze, 3 adet matkap makinesi bulunmaktadır. Ayrıca birer tane punch, vargel ve abkant makinelerinden ve ihtiyaç halinde kullanılmak üzere bir kaynak alanı bulunmaktadır.

Metin 4'e ilişkin elde edilen görsel Şekil 5.16'da verilmiştir.



Şekil 5.15. Metin 3 ile elde edilen model



Şekil 5.16. Metin 4 ile elde edilen model

5.2. Simülasyon

Elde edilen süreç modelleri kullanılarak simülasyon aşamasına gidilebilmektedir. Gerçekleştirilen simülasyon aşaması için bir sözde kod oluşturulmuş ve bu kod Şekil 5.17’ de verilmiştir.

```
Adım 1: Simülasyon verilerini gir.
Adım 2: Her makineye ait dağılım tipini variable2=[] matrisine çek.
Adım 3: Dağılım tiplerine ait parametreleri çek.
Adım 4: Simülasyon süresini çek.
Adım 5: Simülasyon ortamını başlat.
Adım 6: Simülasyon ortamına makine sayısı kadar proses ekle.
Adım 7: Her proses öncesinde ve sonrasında bir kutu oluştur.
Adım 8: Proses için önceki kutudaki parça sayısına bak.
    Eğer kutu boş ise;
        Bir önceki proste işlem bitip kutuya parça konana kadar bekle.
    Değilse;
        Kutudan gereken kadar parça çek.
        Dağılım tipine ve parametrelerine göre imalat süresini belirten bir sayı
        üret.
        Üretilen süre kadar bekle.
        Bir sonraki kutuya bir parça ekle.
Adım 9: Hat veya hücre sonlarında devam eden işlem var ise her noktadan gelecek parça
adedini çek ve Adım 8’e geç.
Adım 10: Simülasyon süresi boyunca Adım 8’i her proses için tekrar et.
Adım 11: Simülasyon çıktıları için bekleyen parça sayısını, üretim süresini ve üretilen
parça sayısını kaydet.
Adım 12: Raporlarda istenilen çıktıları hesapla ve her makineye ait verileri grafiğe
dök.
```

Şekil 5.17. Simülasyon modeline ait algoritma adımları

Şekil 5.17’de verilen sözde kod ile simülasyon aşaması en basit haliyle anlatılmıştır. Burada çekilen simülasyon verileri kullanılarak simülasyon ortamı başlatılır. Ortama makine sayısı kadar proses aşaması eklenir ve her proses öncesi ve sonrasına bir kutu yerleştirilir. Proses için gerekli parça sayısı kutularda yerini alana kadar o proste işlem başlamaz. Yeterli parça halinde üretim süresinin dağılım tipi ve parametrelerine göre bir sayı üretilir. Üretilen sayı proste o an işlem görecekt parça için gerekli süre olarak kabul edilir ve işlem gerçekleşir. Bu durum belirlenen simülasyon süresi kadar devam eder ve

gerekli bilgiler kaydedilir. Kaydedilen bilgiler istenilen çıktıları hesaplamada kullanılır ve elde edilen sonuçlar raporlanır.

Kodun aktif hale gelebilmesi için “Ana sayfa” ara yüzünde sol altta bulunan simülasyon butonuna tıklanır ve simülasyon girdileri için Şekil 5.18’de verilen ara yüz açılır. İlgili görselde 1. metine dair oluşturulan görsel üzerinden simülasyon aşamasına gidilmiştir.

The screenshot shows a software interface for simulation. At the top, there are tabs: "Ana Sayfa", "Simülasyon", "Sonuçlar", and "Veri Analizi". The "Simülasyon" tab is active. Below the tabs, there is a list of machines and their settings. Each machine has a dropdown menu for the distribution type and two input fields for parameter values. The machines are listed in three groups: Group 1 (1. hattaki torna makinesindeki üretim, 1. hattaki freze makinesindeki üretim, 1. hattaki matkap makinesindeki üretim, 1. hattaki vargel makinesindeki üretim, 1. hattaki planya makinesindeki üretim), Group 2 (2. hattaki cnc-torna makinesindeki üretim, 2. hattaki matkap makinesindeki üretim, 2. hattaki punch makinesindeki üretim, 2. hattaki enjeksiyon makinesindeki üretim), and Group 3 (3. hattaki cnc-freze makinesindeki üretim, 3. hattaki planya makinesindeki üretim, 3. hattaki cnc-torna makinesindeki üretim, 3. hattaki abkant makinesindeki üretim). Below the machine settings, there are three input fields for "Hatlardan sırasıyla" (2, 1, 3) and a text field for "adet parça montaj aşamasına gelmektedir.". There are also three input fields for "Montaj aşamasındaki işlem süresi" (5, 10), "Boya aşamasındaki işlem süresi" (9, 2), and "Ambalaj aşamasındaki işlem süresi" (6, 9). At the bottom, there is a text field for "Simülasyon süresi" (150) and a "Tamam" button. The footer of the window says "Eskişehir Teknik Üniversitesi".

Makine	Dağılım Tipi	Parametre Değerleri
1. hattaki torna makinesindeki üretim	Uniform	6, 11
1. hattaki freze makinesindeki üretim	Normal	8, 1
1. hattaki matkap makinesindeki üretim	Uniform	4, 10
1. hattaki vargel makinesindeki üretim	Ussel	6, 0
1. hattaki planya makinesindeki üretim	Normal	9, 2
2. hattaki cnc-torna makinesindeki üretim	Normal	5, 1
2. hattaki matkap makinesindeki üretim	Ussel	5, 0
2. hattaki punch makinesindeki üretim	Uniform	4, 10
2. hattaki enjeksiyon makinesindeki üretim	Ussel	7, 0
3. hattaki cnc-freze makinesindeki üretim	Normal	6, 1
3. hattaki planya makinesindeki üretim	Normal	5, 1
3. hattaki cnc-torna makinesindeki üretim	Normal	6, 2
3. hattaki abkant makinesindeki üretim	Uniform	8, 12

Hatlardan sırasıyla: 2, 1, 3 adet parça montaj aşamasına gelmektedir.

Montaj aşamasındaki işlem süresi: 5, 10

Boya aşamasındaki işlem süresi: 9, 2

Ambalaj aşamasındaki işlem süresi: 6, 9

Simülasyon süresi: 150 dakikadır.

Tamam

Eskişehir Teknik Üniversitesi

Şekil 5.18. Metin 1 ile elde edilen simülasyon ara yüzü

Süreçte yer alan her bir makinenin işlem sürelerine ait dağılım tipi girdileri Şekil 5.18’de görüldüğü gibi 3 dağılım tipinden biri seçilerek belirlenir. Parametrelere ait değerler ise cümle sonlarındaki boşluklara girilir. Seri hat için yapılan bu simülasyon modelinde her hatta tek tip ürün işlendiği varsayılmaktadır. Hat sonlarında işlem görecektür ürünlerin her bir hatta bulunan ürün tiplerinden kaç adet ürün kullanılarak yapılacağı metinde gerekli boşluklara kullanıcı tarafından girilir. En son satırda istenilen simülasyon

süresi dakika cinsinden yazılır ve sol en altta bulunan “Tamam” butonuna basılarak simülasyon başlatılır. Bu simülasyon işlemine dair verilen bilgilerin aynısı U tipi hat modelinin simülasyonu için de geçerlidir. Metin 2 ile elde edilen U tipi simülasyonun girdi ekranı Şekil 5.19’da verilmiştir.

Makine	Üretim Yöntemi	Dağılım	Parametre Değerleri
Torna makinesindeki üretim	Uniform	dağılıma uymaktadır.	5 12
Matkap makinesindeki üretim	Normal	dağılıma uymaktadır.	10 1
Vargel makinesindeki üretim	Normal	dağılıma uymaktadır.	9 2
Freze makinesindeki üretim	Ussel	dağılıma uymaktadır.	6 0
Punch makinesindeki üretim	Uniform	dağılıma uymaktadır.	7 12
Planya makinesindeki üretim	Uniform	dağılıma uymaktadır.	6 10
Abkant makinesindeki üretim	Ussel	dağılıma uymaktadır.	5 0
Hattan sırasıyla	10	adet parça boya aşamasına gelmektedir.	
Boya aşamasındaki işlem süresi	Normal	dağılıma uymaktadır.	8 1
Ambalaj aşamasındaki işlem süresi	Uniform	dağılıma uymaktadır.	4 9
Simülasyon süresi	150	dakikadır.	

Şekil 5.19. Metin 2 modeli ile elde simülasyon ara yüzü

Metin 3 ile elde edilen hücre tipi modelinde sadece ilk iki hücreye ve sonraki işlem adımlarına ait simülasyon ara yüzü oluşturulmuş ve bu ara yüz Şekil 5.20’de verilmiştir. Hücre tipi ve atölye tipi üretim modellerinde, tek bir ürün tipinin üretim sürecine ait bir simülasyon gerçekleştirilmektedir. Bu iki tipte üretilecek ürün U tipi ve seri üretim tipinde olduğu gibi tüm makinelerde işlem görmek zorunda olmadığı için tüm makine işlem sürelerine ait bilgiler girilmemektedir. Yani parçanın işlendiği makineler dışında diğer makinelere ait veri kısımları boş bırakılmaktadır. Bu durum program tarafından işlem görülmeyecek makineler olarak algılanmaktadır. Fakat seri ve U tipi üretimde herhangi bir makine verisinin boş bırakılması durumunda simülasyon gerçekleşmemektedir. Ayrıca makine kullanım sırası da standart olmadığından üretim aşamasında kullanılacak makine sırasına göre makineler seçilir. Hücre ve atölye tipinde aynı makineden birden fazla olması durumu program tarafından algılanmakta ve parça simülasyonda aynı iki makinede zaman geçirmemektedir. Örneğin Şekil 5.20’de 1.

hücrede bulunan iki torna makinesi de seçilmiştir. Bu durum program tarafından herhangi bir kuyruk durumunda iki torna makinesinin de kullanılabilmesi olarak algılanmaktadır.

HATICE Ay - 2019
Değeri Düzenle Hakkında

Ana Sayfa Simülasyon Sonuçlar Yeni Analiz

1 . hücredeki	torna	makinesindeki üretim	Uniform	dağılıma uymaktadır.	Parametre değerleri:	5	10
1 . hücredeki	freze	makinesindeki üretim	Normal	dağılıma uymaktadır.	Parametre değerleri:	8	1
1 . hücredeki	punch	makinesindeki üretim	Usel	dağılıma uymaktadır.	Parametre değerleri:	5	0
1 . hücredeki	torna	makinesindeki üretim	Uniform	dağılıma uymaktadır.	Parametre değerleri:	3	8
1 . hücredeki		makinesindeki üretim		dağılıma uymaktadır.	Parametre değerleri:		
2 . hücredeki	abkant	makinesindeki üretim	Uniform	dağılıma uymaktadır.	Parametre değerleri:	5	10
2 . hücredeki	matkap	makinesindeki üretim	Normal	dağılıma uymaktadır.	Parametre değerleri:	7	1
2 . hücredeki	vargel	makinesindeki üretim	Usel	dağılıma uymaktadır.	Parametre değerleri:	6	0
2 . hücredeki	vargel	makinesindeki üretim	Normal	dağılıma uymaktadır.	Parametre değerleri:	8	2
2 . hücredeki		makinesindeki üretim		dağılıma uymaktadır.	Parametre değerleri:		
2 . hücredeki		makinesindeki üretim		dağılıma uymaktadır.	Parametre değerleri:		

Hücrelerden sırasıyla 2 1 adet parça kaynak aşamasına gelmektedir.

Kaynak aşamasındaki işlem süresi Normal dağılıma uymaktadır. Parametre değerleri: 10 2

Boya aşamasındaki işlem süresi Normal dağılıma uymaktadır. Parametre değerleri: 5 1

Simülasyon süresi 150 dakikadır.

Tamam

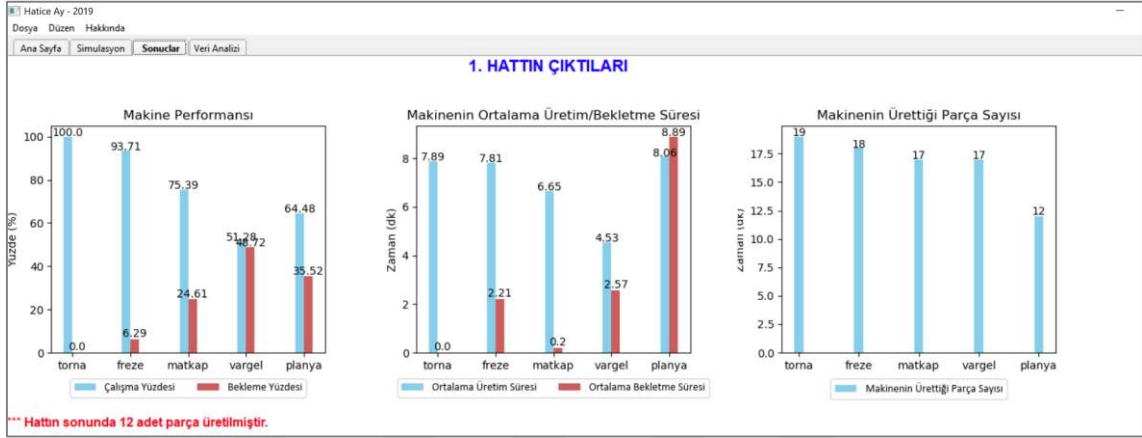
Eskişehir Teknik Üniversitesi

Şekil 5.20. Metin 3 modeli ile elde edilen simülasyon ara yüzü

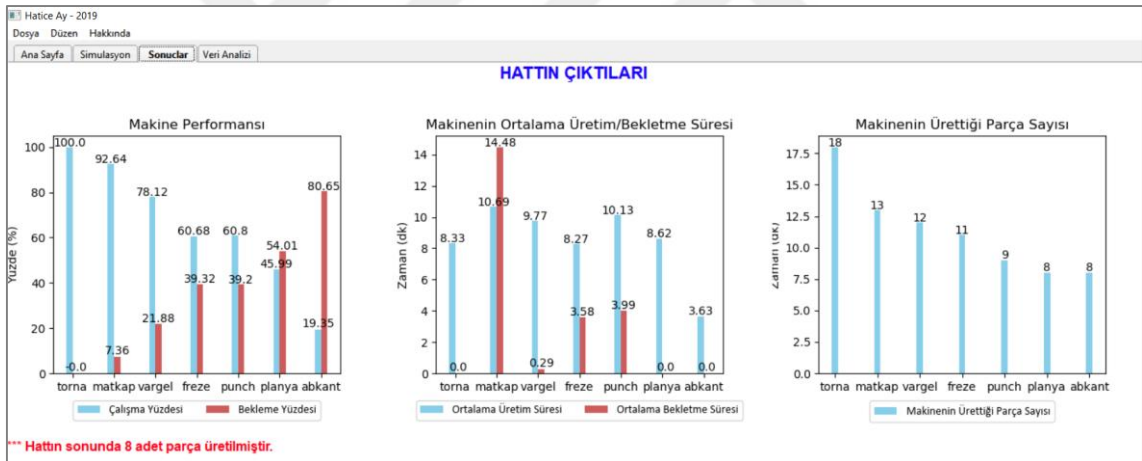
Şekil 5.21’de simülasyon çıktı örneği verilmiştir. Çıktı, Metin 1 ile elde edilen model üzerinden gerçekleştirilen seri üretim sürecinin simülasyonuna aittir. Şekil 5.21’de verilen çıktı örneği her senaryo tipi için aynı şekilde elde edilmektedir. Hücre üretim tipinde her hücreye ve seri üretim tipinde her hatta ait çıktı sonuçları ayrı ayrı verilmektedir. Atölye ve U tipinde tek bir üretim çıktısı yer almaktadır. Şekil 5.22’de U tipinin simülasyon çıktı örneği verilmiştir. Hat ve hücrelerin sonunda ürünler başka bir işleme devam ediyor ise bu sonuçlar hat/hücre çıktılarının en altında yer almaktadır.

Simülasyon çıktılarındaki ilk grafik makine performansına dair bilgileri içermektedir. Yani simülasyon süresi boyunca her makinenin çalışma süresinin yüzdesi hesaplanmaktadır. Hammadde girdisinin stok üzerinden yapıldığı varsayımı sonucu ile ilk makinelerde herhangi bir bekleme yaşanmadan çalışılmaktadır. İkinci grafikte ise parçanın her makine için ortalama işlem süresini ve her makinenin parçayı ortalama kuyrukta bekletme süresi gösterilmektedir. Son grafikte ise simülasyon süresi boyunca

her makinede işlemi tamamlanan parça sayısı verilmektedir. Bu çıktılar sayesinde simülasyon süresi sonunda üretilebilecek ürün sayısı dışında ara stok miktarları ve sistemi yavaşlatan makineler tespit edilebilmektedir.



Şekil 5.21. Metin 1 ile elde edilen modelin simülasyon çıktısı



Şekil 5.22. Metin 2 ile elde edilen modelin simülasyon çıktısı

5.3. Veri Analizi

Simülasyon aşamasında kullanılacak veri setine ait dağılım tipi ve parametre değerleri bilinmiyor ise veri analizi gerçekleştirilerek bu problem giderilebilir. Uygulanan veri analizi için işlem basamaklarını gösteren sözde kod Şekil 5.23'te verilmiştir.

- Adım 1: Analiz edilecek veriyi çek.
- Adım 2: Veriyi virgülden sonra iki basamağa indirge.
- Adım 3: Verinin ortalamasını ve standart sapmasını hesapla.
- Adım 4: Verinin ortalama ve standart sapmasına sahip bir normal dağılım verisi oluştur.
- Adım 5: Verinin ortalamasına sahip bir üssel dağılım verisi oluştur.
- Adım 6: Verinin en küçük ve en büyük değerlerine sahip bir uniform dağılım verisi oluştur.
- Adım 7: Oluşturulan verilere eğri uyumla.
- Adım 8: Uyumlanan eğri ile veri seti karşılaştırarak ki- kare ve Kolmogorov- Smirnov uyum iyiliği testini yap.
- Adım 9: Her dağılım için P- value ve test istatistiği değerlerini hesapla.
- Adım 10: Veri setine ait histogram grafiğini çiz.
- Adım 11: Veri setinin her dağılım için parametreleri ve test sonuçlarını göster.

Şekil 5.23. Veri analizi algoritmasına ait adımlar

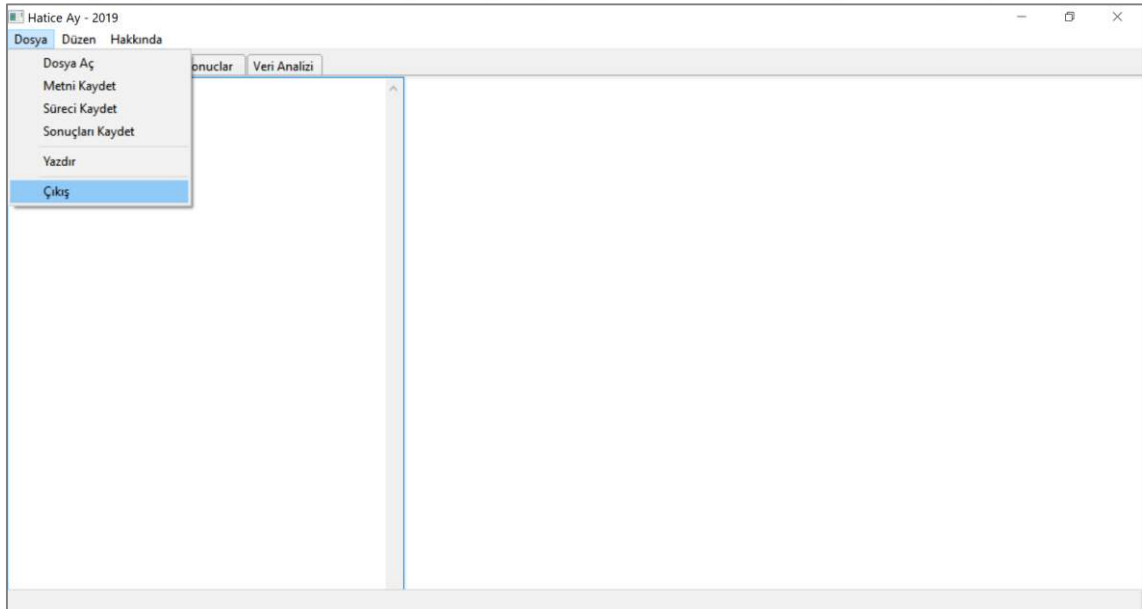
Veri analizi kodu için “Ana Sayfa” ara yüzünde sol altta bulunan “Veri Analizi” kısmı kullanılmalıdır. “Veri Analizi” butonu tıklanarak veriye ait dosya seçilir ve sonuçlar “Veri Analizi” ara yüzünde görüntülenir.

Rastgele oluşturulan bir veri setinin analizi gerçekleştirilmiş ve sonuçlar Şekil 5.24’te verilmiştir. Elde edilen çıktıdaki histogram grafiği veri setindeki değerleri göstermektedir. Ki- kare ve Kolmogorov- Smirnov testleri ile 3 dağılım tipine ait analiz sonuçları verilmiştir. Burada kullanıcı, belirlediği hata payı (α) değerine göre verinin hangi dağılım tipine uyduğunu belirlemektedir. Burada α değeri p - value değerleri ile karşılaştırılmaktadır. Eğer α değeri p - value değerinden küçük ise ($\alpha < p$) bu durumda veri seti p - value değerinin ait olduğu dağılıma uymaktadır yorumu yapılır. Örneğin; Şekil 5.24’teki veri seti sonucunda $\alpha=0.05$ alındığında ki- kare test sonucuna göre bu verinin %95 güven düzeyinde uniform olarak dağıldığı söylenmektedir.



Şekil 5.24. Veri seti analiz sonuçları

Son olarak “Ana Sayfa” ara yüzünde “Dosya” sekmesi altındaki araçların kullanım amaçları açıklanmıştır. Şekil 5.25’te de görüldüğü üzere “Dosya” sekmesinin altında “Dosya Aç”, “Metni Kaydet”, “Süreci Kaydet”, “Sonuçları Kaydet”, “Yazdır” ve “Çıkış” olmak üzere 6 adet araç bulunmaktadır.



Şekil 5.25. Programda kullanılabilecek diğer araçlar

“Dosya Aç” aracı “Ana Sayfa” ara yüzünde sol altta bulunan “Dosya Seç” butonu ile aynı işlevi görmektedir. Süreci oluşturulacak metin dosyası bu araçla da çekilebilmektedir. “Metni Kaydet” aracı kullanılarak süreç modeli için yazılan veya

üzerinde deęişiklikler yapılan metin kaydedilmektedir. “Süreci Kaydet” aracı oluşturulan süreç modelini resim olarak kaydetmeyi, “Sonuçları Kaydet” aracı ise simülasyon çıktılarının kaydedilmesini sağlamaktadır. “Yazdır” aracı ile de hem süreç modeli hem de simülasyon çıktıları yazdırılabilmektedir. “Çıkış” aracı kullanılarak programdan çıkış sağlanmaktadır.



6. SONUÇ VE ÖNERİLER

Gelişen teknoloji ile hayatın her alanında yapay zeka kullanılmaya başlanmış bu durum makinelerin her sektörde yerini almasını sağlamıştır. Bu sayede gün geçtikçe bilgisayar- insan iletişimin artması yapay zekanın alt dalı olan doğal dil işlemeyi bir ihtiyaç haline getirmiştir. Metin veya ses dalgalarının işlenmesi ile makinelerin insanları anlamasını sağlayan doğal dil işleme finans, sanat, sağlık, ulaşım, iletişim gibi birçok sektörde karşımıza çıkmaktadır. Literatürde daha çok metin işlenerek gerçekleştirilen doğal dil işleme uygulamaları bulunmaktadır. Fakat gelişen teknoloji ve insanların makineler tarafından anlaşılma istenmesi, ses işlenerek doğal dil işleme uygulamalarının da artacağını göstermektedir. Bu durum insan hayatını birçok alanda kolaylaştıran sonuçlar doğuracağından hemen hemen her sektörde karşımıza çıkacağı öngörülmektedir.

Yapılan araştırmalardan yola çıkılarak literatürde çok fazla ele alınmayan üretim süreçlerinin doğal dil işlemeyle modellenmesi ve sürece dair simülasyonun gerçekleştirilmesi için, başka bir programa ihtiyaç duyulmadan, sadece sesli anlatım ile tüm sonuçların elde edilmesini sağlayacak bir programın başlangıç niteliğinde bir temelinin oluşturulması amaçlanmış ve bu temel için ilk önce metin işlenerek üretim sürecinin modellenmesi ardından oluşturulan model üzerinden basit bir üretim simülasyonunun gerçekleştirilmesi önerilmiştir.

Gerçekleştirilen çalışma Python 3.7 kullanılarak kodlanmıştır. Amaç doğrultusunda yazılan program ile üretim sürecine dair alınan metin verisi işlenerek aranan bilgiler çekilmeye çalışılmıştır. Metin işleme tabanlı oluşturulan model ile veriden üretim sürecinin tipine, süreçte kullanılan makinelere, makinelerin konumlarına ve sayılarına dair bilgiler çekilmiştir. Daha sonra çekilen bilgiler görselleştirilerek süreç modellenmiştir.

Modellenen süreç şekline göre makine işlem sürelerine dair gerekli bilgiler kullanıcıdan alınarak simülasyon işlemi gerçekleştirilmiş ve çıktılar birkaç farklı şekilde yorumlanmıştır. Ayrıca simülasyon aşamasında veri setinin dağılım tipinin ve parametrelerinin belirlenmesine dair iki farklı yöntemle veri analizi gerçekleştirilmiştir.

Çalışma sonucunda çok kısa sürede üretim sürecini modelleyebilen bir program elde edilmiştir. Ayrıca bu alanda benzer bir uygulama olmaması çalışmanın literatüre özgün bir katkı sağlayacağını düşündürmektedir.

Çalışmanın bir sonraki adımında girdinin metin yerine sesli olarak verilmesi gerçekleştirilebilir hatta karşılıklı etkileşimli bir süreç oluşturulabilir. Programın

geliştirilebilecek en büyük yanı ise simülasyon kısmıdır. Çalışmanın simülasyon kısmında makine arızalarına ve aynı anda birden fazla ürün tipi üretilmesine yer verilmemiştir. İlerleyen çalışmalarda bu kısma yer verilebilir. Ayrıca hücre tipi imalat kısmında makine- parça matrisi kullanılarak hücreler ve makinelerin hangi hücrede olacağı program tarafından oluşturulabilir. Hatta bu matris kullanılarak parçaların akış şekli çekilerek her ürün tipi için üretim süreci tanımlanabilir.



KAYNAKÇA

- Adalı, E. (2012). Doğal dil işleme. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 5(2).
- Bahl, L.R., Jelinek, F. ve Mercer, R.L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 5(2), 179-190.
- Bird, S., Loper, E. ve Klein, E. (2009). *Natural language processing with Python*, USA: O'Reilly Media Inc.
- Brown, P.S., Cocke, J., Pietra, S.A., Pietra, V.C., Jelinek, F., Lafferty, J.D., Mercer, R.L. ve Roossin, P.S. (1990). A statistical approach to machine translation. *Computational Linguistic*, 16(2), 79-85.
- Chen, L., Song, L., Shao, Y., Li, D. ve Ding, K. (2019). Using natural language processing to extract clinically useful information from Chinese electronic medical records. *International Journal of Medical Informatics*, 124, 6-12.
- Cheng, Y., Jia, Y., Fang, R., She, L., Xi, N. ve Chai, J. (2014). Modelling and analysis of natural language controlled robotic systems. *IFAC Proceedings Volumes*, 47(3), 11762-11772.
- Demirdöğen, O. (1998). Talep tahmininde Monte Carlo simülasyon tekniğinin kullanılması. *Atatürk Üniversitesi İİBF Dergisi*, 12 (1-2), 229-240.
- Dempster, A.P., Laird, N.M. ve Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1), 1-38.
- Erimçağ, H. (1986). Satış tahmin metodu olarak Monte-Carlo simülasyonu. *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 15(1), 132.
- Hindle, D. ve Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1), 103-120.
- Hunter, J.D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95.
- Kartal, M. (1998). *Hipotez testleri*. Erzurum: Şafak Yayınevi.
- Large, D.R., Clark, L., Quandt, A., Burnett, G. ve Skrypchuk (2017). Steering the conversation: A linguistic exploration of natural language interactions with a digital assistant during simulated driving. *Applied Ergonomics*, 63, 53-61.

- Manns, M., Wallis, R. ve Deuse, J. (2015). Automatic proposal of assembly work plans with a controlled natural language. *Procedia CIRP*, 33, 345-350.
- Matci, D. ve Avdan, U. (2018). Address standardization using the natural language process for improving geocoding results. *Computers, Environment and Urban System*, 70, 1-8.
- Mikolov, T., Corrado, G., Chen, K. ve Dean, J. (2013). Efficient estimation of word representations in vector space. *Proceedings of the International Conference on Learning Representations* sunulan bildirir. Scottsdale, Arizona.
- Nivre, J. ve Scholz, M. (2004). Deterministic dependency parsing of English text. *Proceedings of the 20th international conference on Computational Linguistics*, Geneva, Switzerland.
- Oliphant, T.E (2006). *A guide to NumPy*. USA: Trelgol Publishing.
- Reshamwala, A., Mishra, D. ve Pawar, P. (2013). Review on natural language processing. *Engineering Science and Technology: An International Journal*, 3(1), 113-116.
- Swartz, j., Koziatsek, C., Theobald, J., Smith, S. ve Iturrate, E. (2017). Creation of a simple natural language processing tool to support an imaging utilization quality dashboard. *International Journal of Medical Informatics*, 101, 93-99.
- Şeker, Ş.E. (2015). Doğal dil işleme (natural language processing). *YBS Ansiklopedi*, 2(4).
- Taha, H. (2014). *Yöneylem araştırması*. (Çev: Ş.A. Baray ve Ş. Esnaf). İstanbul: Literatür Yayıncılık.
- Tan, W.K., Hassanpour, S., Heagerty, P.J., Rundell, S.D., Suri, P., Huhdanpaa, H.T., James, K., Carrell, D.S., Langlotz, C.P., Organ, N.L., Meier, E.N., Sherman, K.J., Kallmes, D.F., Luetmer, P.H., Griffith, B., Nerenz, D.R. ve Jarvik, J.G. (2018). Comparison of natural language processing rules-based and machine-learning systems to identify lumbar spine imaging findings related to low back pain. *Academic Radiology*, 25(11), 1422-1432.
- Tanguy, L., Tulechki, N., Urieli, A., Hermann, E. ve Raynal, C. (2015). Natural language processing for aviation safety reports: From classification to interactive analysis. *Computers in Industry*, 78, 80-95.
- Wachtel, A., Weigelt, S. ve Tichy, W. F. (2016). Initial implementation of natural language turn-based dialog system. *Procedia Computer Science*, 84, 49-56.
- Wachter, M., Ovchinnikova, E., Wittenbeck, V., Kaiser, P., Szedmak, S. Mustafa, W., Kraft, D., Krüger, N., Piater, J. ve Asfour, T. (2018). Integrating multi-purpose

- natural language understanding, robot's memory, and symbolic planning for task execution in humanoid robots. *Robotics and Autonomous Systems*, 99, 148-165.
- Wichmann, P., Brintrup, A., Baker, S., Woodall, P. ve McFarlane, D. (2018). Towards automatically generating supply chain maps from natural language text. *IFAC PapersOnLine*, 51(11), 1726-1731.
- Woods, W.A. (1970). Transition network grammars for natural language analysis. *Computational Linguistics*, 13(10), 591-606.
- van der Walt, S., S. Chris Colbert, S.C. ve Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13, 22-30.
- Van Le, D., Montgomery, J., Kirkby, K.C. ve Scanlan, J. (2018). Risk prediction using natural language processing of electronic mental health records in an inpatient forensic psychiatry setting, *Journal of Biomedical Informatics*, 86, 49-58.
- Zhang, F., Fleyeh, H., Wang, X. ve Lu, M. (2019). Construction site accident analysis using text mining and natural language processing techniques. *Automation in Construction*, 99, 238-248.
- Zou, Y., Kiviniemi, A. ve Jones, S.W. (2017). Retrieving similar cases for construction Project risk management using Natural Language Processing techniques. *Automation in Construction*, 80, 66-76.
- http-1:** <https://www.kursunkalem.com/gramer-terimi/dil-bilimi/>
(Erişim tarihi: 04.06.2019)
- http-2:** <https://medium.com/t%C3%BCrkiye/do%C4%9Fal-dil-i%C4%B1%87%C5%9Fleme-rehberi-2c4c41260f74> (Erişim tarihi: 04.06.2019)
- http-3:** <https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908> (Erişim tarihi: 12.05.2019)
- http-4:** <https://wxpython.org/> (Erişim tarihi: 08.06.2019)
- http-5:** <https://github.com/otuncelli/turkish-stemmer-python> (Erişim tarihi: 28.05.2019)
- http-6:** Jones, E., Oliphant, E. ve Peterson, P. (2001). SciPy: Open Source Scientific Tools for Python: <http://www.scipy.org> (Erişim tarihi: 20.06.2019)
- http-7:** <https://simpy.readthedocs.io/en/latest/> (Erişim tarihi: 17.06.2019)

ÖZGEÇMİŞ

Adı Soyadı : Hatice AY
Yabancı Dil : İngilizce
Doğum Yeri/ Yılı : Edirne/ 1994
E- Posta : htc.ay1001@gmail.com

Eğitim Geçmişi:

- 2017, Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği, Endüstri Mühendisliği Anabilim Dalı

Yayınlar:

- 2017, Özet Bildiri, An efficient resource planning software proposal for Small Medium Enterprises (SMEs): A case study in a factory, *International Conference on Computational and Statistical Methods in Applied Sciences*, Samsun: On Dokuz Mayıs Üniversitesi, s.14.