ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Soner TÜRKOĞLU

DEVELOPING OXYGEN AMOUNT PREDICTION MODEL OF BASIC OXYGEN FURNACE STEELMAKING PROCESS WITH MACHINE LEARNING ALGORITHMS

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2017

ABSTRACT

MSc. THESIS

DEVELOPING OXYGEN AMOUNT PREDICTION MODEL OF BASIC OXYGEN FURNACE STEELMAKING PROCESS WITH MACHINE LEARNING ALGORITHMS

Soner TÜRKOĞLU

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Asst. Prof. Dr. Buse Melis ÖZYILDIRIM
Year: 2017, Pages: 41
Jury : Asst. Prof. Dr. Buse Melis ÖZYILDIRIM
Assoc. Prof. Dr. Umut ORHAN
Asst. Prof. Dr. Fatih KILIÇ

Basic Oxygen Furnace (BOF) steelmaking is a highly complex multiphase physics and chemical reaction process, which aims to reduce the carbon content and increase temperature. During the process amount of oxygen blown into BOF is the most significant control parameter, which effects life of BOF refractory lining, production cost and producing steel in desired quality. In this thesis, the purpose is developing oxygen amount prediction of dynamic blow stage in BOF steelmaking process with machine learning algorithms. For finding the best prediction model, Linear Regression, Multi-Layer Perceptron (MLP), K-Nearest Neighbor (KNN), Extreme Learning Machine (ELM) and Support Vector Machine (SVM) based Sequential Minimal Optimization (SMO) Regression algorithms were compared using Weka and MATLAB. The results of the prediction models developed with gathered production data from BOF are close to each other. ELM and SVM based SMO regression predicted slightly better results than the other models.

Key Words: Weka, Matlab, Machine Learning, Basic Oxygen Furnace, oxygen volume prediction

ÖZ

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE BAZİK OKSİJEN FIRINI ÇELİK ÜRETİMİNDE OKSİJEN MİKTARI TAHMİNLEME MODELİ GELİŞTİRİLMESİ

Soner TÜRKOĞLU

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Yrd. Doç. Dr. Buse Melis ÖZYILDIRIM
Yıl: 2017, Sayfa: 41
Jüri : Yrd. Doç. Dr. Buse Melis ÖZYILDIRIM
Doç. Dr. Umut ORHAN
Yrd. Doç. Dr. Fatih KILIÇ

Bazik Oksijen Fırını (BOF) ile çelik üretimi, karbon içeriğini düşürmeyi ve sıcaklığı yükseltmeyi amaçlayan oldukça karmaşık, birçok fazlı fizik ve kimyasal reaksiyon içeren bir işlemdir. Proses esnasında BOF içerisine üflenen oksijen miktarı BOF refrakter kaplamasının ömrünü, üretim maliyetlerini ve istenen kalitede çelik üretimi yapılabilmesini etkileyen en önemli kontrol parametresidir. Bu tezde amaç BOF ile çelik üretimi sürecinde dinamik üfleme aşamasında üflenen oksijen miktarını makine öğrenmesi algoritmaları ile tahminleme modelinin geliştirilmesidir. En iyi tahminleme modelinin bulunabilmesi için, Doğrusal Regresyon, Çok katmanlı Algılayıcı(MLP), K-En Yakın Komşu (KNN), Aşırı Öğrenme Makinesi (ELM) ve Destek Vektör Makineleri (SVM) tabanlı Sıralı Minimal Optimizasyon (SMO) Regresyonu algoritmalarının Weka ve MATLAB kullanılarak karşılaştırılması yapılmıştır. BOF üretiminden toplanan verilerle geliştirilen tahminleme modellerinin sonuçları birbirlerine çok yakın çıkmıştır. ELM ve SVM tabanlı SMO Regresyonu ile diğer modellerden biraz daha iyi sonuç tahminleme yapılmıştır.

Anahtar Kelimeler: Weka, Matlab, Makine Öğrenmesi, Bazik Oksijen Fırını, oksijen miktarı tahmini

GENİŞLETİLMİŞ ÖZET

Ham Çelik üretimi tüm dünyada yıllar geçtikçe artmaktadır. 2000 yılında 850 milyon ton ham çelik üretildi. Üretim rakamları 15 yılda neredeyse iki katına çıktı ve 2015 yılına gelindiğinde dünyadaki ham çelik üretimi 1621 milyon ton oldu. Tüm dünyayı etkisi altına alan ekonomik krizden dolayı 2009 yılında küresel üretim düştü. Ancak 2010 yılında tekrardan toparlandı ve üretimde yükseliş eğilimi devam etti. 2014 yılında 1670 milyon ton üretilen ham çelik miktarı 2014 ten 2015 yılına geçişte yüzde 2.8 düşüş gösterdi.

Türkiye'de ham çelik üretimi iki yöntemle yapılmaktadır. Bunlardan ilki elektrik ark ocaklarıyla yapılan üretimdi. Elektrik ark ocaklarıyla üretimde hammadde olarak ağırlıklı olarak hurda kullanılır. Diğeri ise demir cevherinden elde edilen sıvı ham demire oksijen üflenerek yapılan bazik oksijen fırınlı üretimdir. 2015 içerisindeki 11.04 milyon ton ham çelik üretiminin yüzde 35'i bazik oksijen fırınlarıyla yapılmıştır.

BOF ile ham çelik üreten bir demir çelik fabrikasının üretim akışı, fabrikaya iki temel malzeme olan kömür ve cevherin girişiyle başlar. Kömür, kok bataryaları kullanılarak kok üretiminde kullanılır. Demir cevheri için ise ilk durak sinterleme işlemi için sinter fabrikalarıdır. Kok, sinter ve diğer demir cevheri malzemeleri sıvı ham demir üretimi için yüksek fırınlarda kullanılır. Üretilen sıvı ham demir hurda, kireç ve oksijen kullanılarak bazik oksijen fırınlarında sıvı çeliğe dönüştürülür. Sıvı çeliğin istenilen kalite değerlerine ulaşabilmesi için pota fırınlarında içerisine alaşım elementleri eklenir. Sıvı çelik formunun son durağı ise sürekli döküm tesisleridir. Bu tesislerde sıvı çelik slab ve kütük formuna sokulur.

BOF prosesi fırına hurda eklenmesi ile başlar. Hurda, demir içerisindeki çözünmüş elementlerin oksidasyonu ile açığa çıkan ısının etkisinin azaltılması için soğutucu olarak kullanılır. Kükürt giderme işlemi sonrasında sıvı ham demir, hurdanın ardından BOF içerisine aktarılır. Oksijen lansı banyo seviyesi (hurda ve sıvı ham demirin fırın tabanından yüksekliği) üzerine kadar gelerek yüksek

basınçla oksijen üflemeye başlar. Bununla birlikte ek malzemeler fırının üzerinden fırın içerisine aktarılır. Toplam üflenen oksijen 7000 – 7500 Nm^3 seviyelerine geldiğinde sub-lans sıcak metalin içerisine TSC (Sıcaklık Numune Karbon) başlığıyla iner ve sıvı metalin karbon ve sıcaklığını ölçerek aynı zamanda belirli bir miktar numune alır. Bu ölçüm sıvı metalin içeriği hakkında bilgi verir. Bu ölçüm baz alınarak daha ne kadar oksijen üfleneceği ve katkı malzemesi ekleneceği belirlenir. Oksijen üflemeye başlamasından TSC ölçümü alınmasına kadar ki geçen süreye ana üfleme denir. TSC ölçümü alındıktan sonra üflemeye devam edilir ve bu sürece de dinamik üfleme süreci denir. Dinamik üfleme süreci TSO (Sıcaklık Numune Oksijen) başlığıyla ölçüm alınmasına kadar devam eder. TSO başlığıyla sıvı çeliğin sıcaklığı oksijen değeri ölçülür ve aynı zamanda analizinin yapılması amacıyla laboratuvara gönderilmek üzere numune alınır. Eğer TSO ölçüm sonuçları sıvı çeliğin istenilen kalite değerlerini karşılaması durumunda döküm çelik potasına aktarılarak tamamlanır. İstenen kalite değerlerine ulaşılamamışsa tekrar üfleme aşaması başlar.

Min Han ve Yao Zhao (2011) yaptıkları bir çalışmada adaptif ağ tabanlı bulanık mantık çıkarım sistemi ile dinamik üfleme adımındaki oksijen miktarının tahminini yapmışlardır. Bu çalışmada giriş verisi olarak ana üfleme sonunda alınan karbon ve sıcaklık değerleri seçilmiş. Eğitim için 300 test için 100 veri seti kullanılmış. Min Han ve arkadaşlarının(2014) daha önce yaptıkları bir çalışmadaki verilerle karşılaştırma yaparak kullanılan yöntemin daha iyi olduğunu ortalama hata karelerinin karekökü değerini 170.45 m^3 bularak belirtmişler. Aynı zamanda bir diğer değerlendirme parametresi olarak hedef değere ulaşma oranı seçilmiş ve eşik değer ±200 m^3 seçilerek yüzde 86 bulunmuş.

Bu çalışmamızda dinamik üfleme esnasındaki oksijen miktarı Lineer Regresyon, K En Yakın Komşu, Destek Vektör Makineleri, Radyal Tabanlı Fonksiyon Ağı, Çok Katmanlı Yapay Sinir Ağı ve Uç Öğrenme Makineleri yöntemleri kullanılarak tahmin edilmiştir. Veriler 200 tonluk BOF çelik üretiminden alınmıştır. Eğitim ve test için toplamda 3988 veri seti kullanılmıştır.

Hedef karbon ve sıcaklık değerleri, TSC karbon ve sıcaklık değerleri, sıvı ham demir ve hurda ağırlıkları ve banyo seviyesi giriş verileri olarak seçilmiştir.

Tahminleme, Weka ve Matlab araçları kullanılarak yapılmıştır. Toplamda 5 değerlendirme parametresi kullanılmıştır. Bunlar korelasyon katsayısı, ortalama hata karelerinin karekökü, ortalama mutlak hata, bağıl mutlak hata ve bağıl hata karelerinin kareköküdür. Modellerin pratikte ne kadar doğrulukta çalıştığını bulabilmek için 10 katmanlı çapraz doğrulama yöntemi kullanılmıştır. Parametrelerde uygun değerleri bulabilmek için ise örgü arama yöntemi kullanılmıştır.

Weka kullanılarak yapılan tahminleme modellerinde sonuçlar birbirine yakın çıkmıştır. En iyi sonuç destek vektör makineleri tabanlı ardışık minimal optimizasyon regresyon yöntemi ile bulunmuştur. Radial Tabanlı fonksiyon kullanılarak yapılan denemelerde en iyi sonucu verecek C parametresi $[10^{-3}, 10^{3}]$, gamma parametresi $[10^{-3}, 10^{2}]$ aralıklarında örgü arama yöntemi ile aranmıştır. C 15 ve gamma 0.13 değerlerinde; korelasyon katsayısı 0.785, ortalama hata karelerinin karekökü 183.875, ortalama mutlak hata 141.771, bağıl mutlak hata 0.594 ve bağıl hata karelerinin karekökü 0.620 ile en iyi performans değerleri bulunmuştur.

Uç öğrenme makinesi metodu da eklenerek aynı veriler Matlab kullanılarak da tahminleme yapılmıştır. Aynı değerlendirme parametreleri ışığında Weka ile yaklaşık aynı sonuçlar elde edilmiştir. Yeni eklenen uç öğrenme makinesi yöntemi ile yapılan tahminin modelinin korelasyon katsayısı destek vektör makineleri tabanlı ardışık minimal optimizasyon regresyon yöntemi ile aynı çıkmıştır.

BOF kullanılarak demirden çeliğe dönüşümün en önemli aşaması oksijen üfleme safhasıdır. Üflenen oksijen miktarı BOF içerisinde kaplı olan refrakter ömrüne, üretim maliyetlerine ve üretilecek çeliğin istenen kalitede olmasına çok büyük etkisi vardır. Makine öğrenmesi algoritmaları kullanılarak yapılan bu çalışmada Matlab ve Weka karşılaştırması da yapılmıştır. Weka kullanım kolaylığı

olarak Matlab uygulamasının önüne geçse de algoritma parametrelerinin yazılımdaki seçenek kısıtlamaları uygulamanın dezavantajı olarak belirlenmiştir.

İleride yapılacak çalışmalarda BOF işlem sürecini etkileyecek farklı etkenler giriş parametresi olarak seçilebilir. Girişlerin artması ile giriş değerleri seçme yöntemleri kullanılarak en uygun girişler seçilebilir. Aynı zamanda son yıllarda algoritmaların birleştirilmesi ile oluşturulan yeni yöntemler veri setine uygulanabilir.

ACKNOWLEDGEMENTS

I would first like express my sincere appreciation to the constant support and encouragement of my advisor Asst. Prof. Dr. Buse Melis ÖZYILDIRIM during this study.

My sincere thanks to Asst. Prof. Dr. Çiğdem ACI and Asst. Prof. Dr. Mehmet ACI for their support and motivation.

I would also thank my friends Ümit Abdullah TANRIKULU, Önder Ziya PAYLAR, Görkem Turan ESKİYURT and my colleagues at work who provided valuable input and motivation during the completion of my thesis.

Finally special thanks to my family for their understandings and patience.

CONTENTS **PAGE**

ABSTRACT I
ÖZ II
GENİŞLETİLMİŞ ÖZET III
ACKNOWLEDGEMENTS VII
CONTENTS VIII
LIST OF TABLES X
LIST OF FIGURES XII
LIST OF ABBREVIATIONS XIV
1. INTRODUCTION 1
1.1. Global Crude Steel Production 1
1.2. Production Flow in an Iron and Steel Factory 4
1.3. BOF Process 5
1.4. Chemical Reactions in BOF Steelmaking 9
2. PREVIOUS WORKS 11
3. METHODS 15
3.1. Linear Regression 15
3.2. Multi-Layer Perceptron 16
3.3. Radial Basis Function Network 18
3.4. Extreme Learning Machine 20
3.5. K-Nearest Neighbor Algorithm 23
3.6. Support Vector Machine 26
4. RESULTS AND DISCUSSIONS 29
5. CONCLUSION 35
REFERENCES 37
CURRICULUM VITAE 41

LIST OF TABLES **PAGE**

Table 1.1. 2014 and 2015 crude steel production (World Steel in Figures, 2016)…… 3

Table 1.2. Erdemir Group's crude steel production and rank in the world……..... 4

Table 1.3. Chemical reactions occurring in a BOF (Brooks et al,2002)..... 9

Table 2.1. Prediction model results of the oxygen blowing volume calculation (Min Han et al, 2014)…….. 12

Table 2.2. Comparison of CBR and ANFIS of the study (Han and Zhao, 2011)…….. 12

Table 4.1. Description of variables used in BOF steelmaking…… 29

Table 4.2. Weka results……..... 31

Table 4.3. Matlab results……. 33

LIST OF FIGURES **PAGE**

Figure 1.1.	Global crude steel production by years……..	1
Figure 1.2.	Top ten crude steel producing countries (Global Steel Report, 2016)…….	2
Figure 1.3.	Crude steel production in Turkey over years ……	3
Figure 1.4.	Production flow …….	5
Figure 1.5.	Schematic view of a BOF……..	6
Figure 1.6.	Schematic of operational steps in BOF (Miller et al.,1998)...	8
Figure 1.7.	Graphic of oxygen blown to BOF and process phases of a heat…….	8
Figure 2.1.	Prediction results of the (Han and Zhao,2011) study…….	13
Figure 2.2.	Relative errors of the models of the Wang et al (2012) study	14
Figure 3.1.	Representation of MLP……...	17
Figure 3.2.	Representation of RBF……...	19

LIST OF ABBREVIATIONS

BOF	: Basic Oxygen Furnace
TSC	: Temperature Sample Carbon
TSO	: Temperature Sample Oxygen
SVM	: Support Vector Machine
CBR	: Case-Based Reasoning
DE CBR	: Decarburization efficiency added Case-Based Reasoning
DE I-CBR	: Decarburization efficiency added Mutual Information and Case-Based Reasoning
RMSE	: Root Mean Square Error
HR	: Hit Rate
ANFIS	: Adaptive-Network-Based Fuzzy Inference System
ANN	: Artificial Neural Network
MLP	: Multi-Layer Perceptron
RBF	: Radial Basis Function
SLFN	: Single Layer Feed Forward Networks
ELM	: Extreme Learning Machine
KNN	: K-Nearest Neighbor
GD	: Gradient Descent
MLR	: Multivariate Linear Regression
CC	: Correlation Coefficient
RRSE	: Root Relative Square Error
MAE	: Mean Absolute Error
RAE	: Relative Absolute Error

1. INTRODUCTION

Steel plays a key role in the new circular business model in which all products must be designed and manufactured to be repaired, remanufactured, reused and recycled (World Steel in Figures, 2016). The next chapters include information about global steel production and how steel is produced.

1.1. Global Crude Steel Production

Crude steel production has been growing over the years in the whole world. 850 million tons crude steel produced in 2000 and production had grown 771 million tons. Production figures have almost doubled in 15 years and in 2015 crude steel production was 1621 million tons. Due of the financial crisis, global production has decreased in 2009 but recovered by 2010. World crude steel reached the max production level in 2014, 1670 million tons but in 2015 because of the low demand, it fell 2.8 percent from 2014 and was 1621 million tons. (World Steel in Figures, 2016)

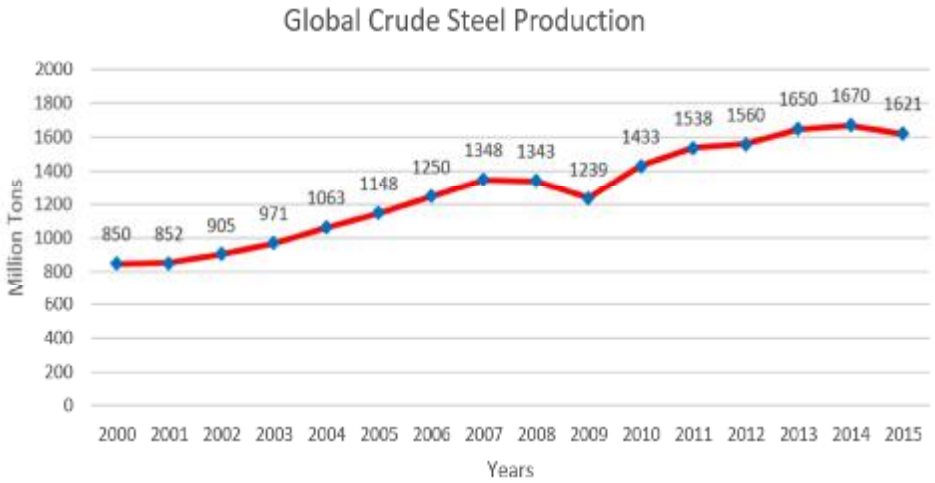

Figure 1.1. Global crude steel production by years

China, Japan, India, US, Russia, South Korea, Germany, Brazil, Turkey, and Ukraine are the top steel producing countries, respectively.

Figure 1.2.Top ten crude steel producing countries (Global Steel Report, 2016)

China produced 803.83 million tons of crude steel in 2015; it is 49.5 percent of the world's total production. Most of the steel producing companies in China is belong to the state. The rapid development of China has brought the most use of its own steel production (World Steel in Figures, 2016)

China's steel prices are very cheap and extremely competitive in the world market. The reason for this is large-scale production. Many countries have attempted to impose an import tax on steel imports from China in a bid to encourage consumption of locally made steel.

In 2015, Turkey's crude steel production capacity grew slightly by 0.5% from 50.2 million tons to 50.4 million tons, which shows that growth pace slowed substantially compared to recent years (Turkish Steel Industry, 2015).

Turkey ranked 8th in 2014, with 34 million tons of crude steel production. But in 2015 the production fell and ranked 9th after Brazil with 31.5 million tons.

Table 1.1. 2014 and 2015 crude steel production (World Steel in Figures, 2016)

Country	2015		2014	
	Rank	Tonnage (Million tons)	Rank	Tonnage (Million tons)
China	1	803.8	1	822.8
Japan	2	105.2	2	110.7
India	3	89.4	4	87.3
United States	4	78.8	3	88.2
Russia	5	70.9	6	71.5
South Korea	6	69.7	5	71.5
Germany	7	42.7	7	42.9
Brazil	8	33.3	9	33.9
Turkey	9	31.5	8	34.0
Ukraine	10	23.0	10	27.2
Italy	11	22.0	11	23.7
Taiwan, China	12	21.4	12	23.1
Mexico	13	18.2	13	18.9
Iran	14	16.1	14	16.3
France	15	15.0	15	16.1

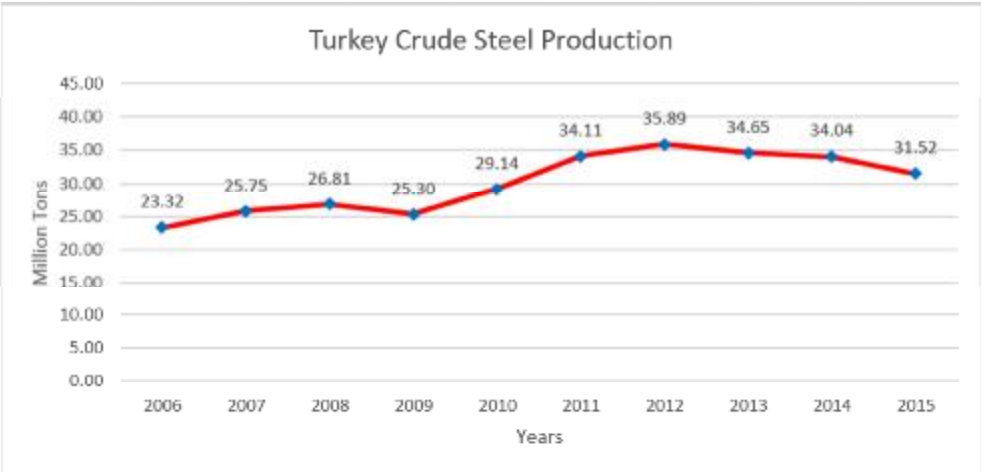

Figure 1.3. Crude steel production in Turkey over years

There are two types of crude steel production in Turkey. One of them is with electric arc furnace that rely heavily on scrap and the other is with oxygen blown converters. Oxygen blown converters converts hot metal to crude steel with oxygen. In 2015, 11.04 million tons crude steel, 35 percent of total production, produced with oxygen blown converters. Rest of the crude steel production made by electric arc furnaces (Steel Statistical Yearbook, 2016).

Three iron and steel company in Turkey is producing crude steel with oxygen blown converters: Kardemir, Erdemir and Isdemir. Erdemir and Isdemir are belong the same group, Erdemir Group. Erdemir Group is the biggest industrial corporation of Turkey in terms of total assets, with 7 companies. Group is one of the major players in iron and steel industry with industrial facilities in Turkey and Romania and owns 80% of iron ore reserves in Turkey. In 2015 with 8.9 million tons crude steel production ranked 43th in the World and 8th in Europe countries.

Table 1.2. Erdemir Group's crude steel production and rank in the world.

World	Europe	EU-28	Company	2015 Production (million tons)
1	1	1	ArcelorMittal (International)	97.1
16	2	2	TyssenKrupp (Germany)	17.3
43	***8***	***3***	***Erdemir Group***	***8.9***
48	9	4	Voestalpine (Austria)	7.8

1.2. Production Flow in an Iron and Steel Factory

Production flow of an iron and steel factory producing crude steel with BOF can be seen on Figure 1.4. Coal is used for production of coke at the coke ovens and iron ore is used at sinter plant. Coke, sinter and other iron ore materials such as pellets are used for the production of hot metal at the blast furnaces. The

hot metal is reduced to liquid steel at BOF by using scrap, limestone and oxygen. Also at the secondary metallurgy, (ladle furnace) alloying elements are added to liquid steel to obtain the desired quality.

The liquid steel is converted into slab or billet at the continuous casting plant: slab casting and billet casting units. Wire rod is produced from the billets at the wire rod mill and hot rolled coil is produced from slabs at the hot strip mill.

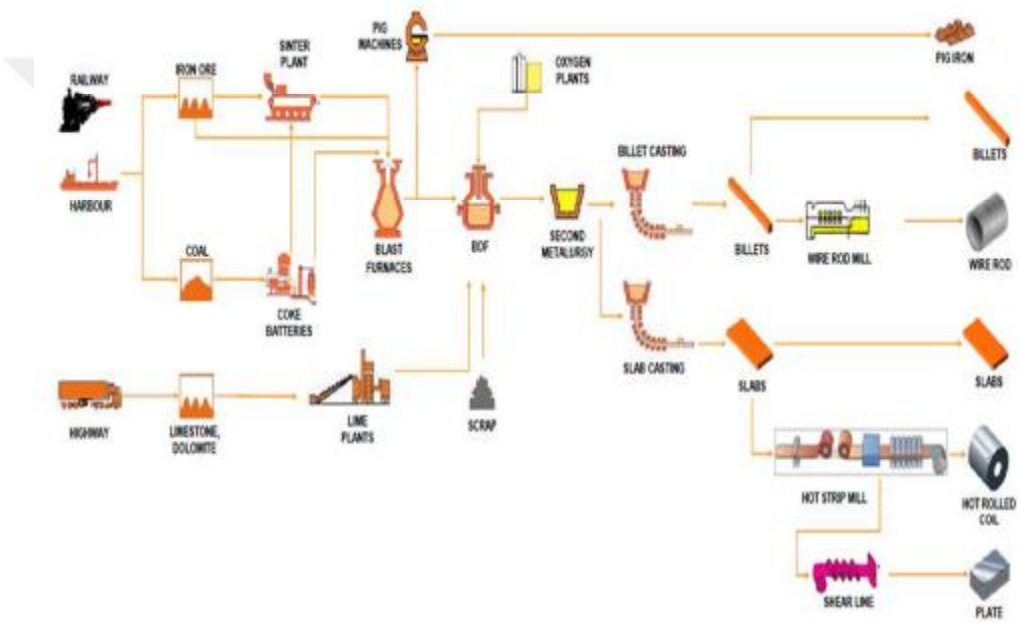

Figure 1.4. Production flow

1.3. BOF Process

One of the main process of iron and steel production is BOF process. It is a typical complex batch chemical reactor with sophisticated thermal and chemical reaction processes, which convert liquid pig iron into steel with desired grade (temperature and chemical compositions, such as carbon, etc.) under oxidation conditions (Chen et al.,2010).

The goal of BOF process is converting molten iron and scrap into steel of desired carbon and temperature using oxygen. Before BOF process, molten iron poured into a ladle from torpedo and transferred to desulphurization plant by cranes. At desulphurization plant the goal is remove the Sulphur from hot metal content to reach target sulphur by injecting magnesium and lime. Target Sulphur is determined by the steel grade selection going to produce at BOF. Every ladle full of hot metal is discrete batches and called heats. A heat starts from desulphurization plant and ends at casting plant.

Schematic view of a top blowing BOF can be seen on Figure 1.5. Inner part of BOF is refractory lined and the shell is steel. At the top waste gas collector system, collects the waste gas generated during the chemical reactions and materials are also added to process from the top by material handling system. On the side wall of BOF there is a tap hole for tapping steel to teeming ladles. Oxygen blown into BOF by the water-cooled oxygen lance. Oxygen lance can move up or down by commands during the operation according to process conditions.

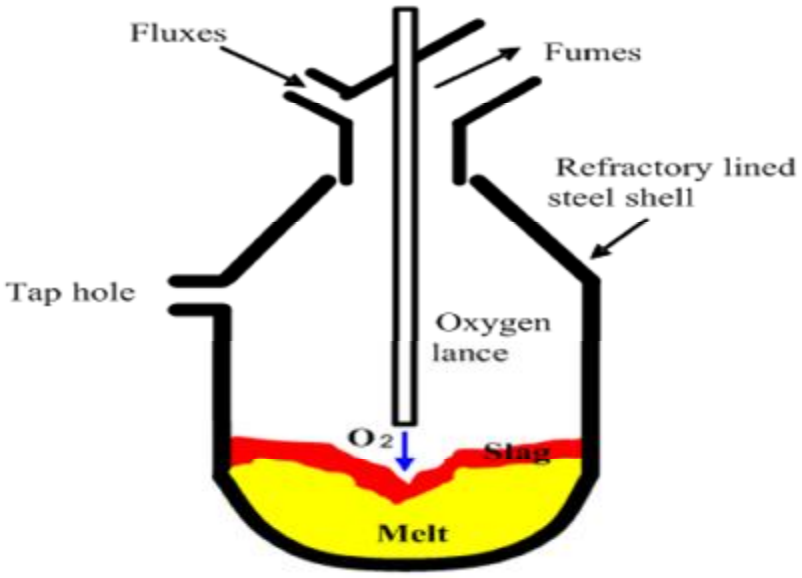

Figure 1.5. Schematic view of a BOF

BOF process starts with charging scrap into the furnace. Scrap is used as a coolant for the heat generated by the oxidation of elements dissolved in iron. After desulphurization treatment hot metal charges to the furnace. Oxygen lance move downwards above the bath level (hot metal and scrap mixture height) and begin to blow oxygen with high pressure. In addition, flux materials starts to charge into the furnace by material handling system. When total blown oxygen is reached 7000 – 7500 Nm3 sub-lance dives into the hot metal and take a sample. This sample is taken with TSC probe. TSC stands for Temperature Sample Carbon. With this probe type, temperature and carbon are measured and sample is sent to the laboratory to analyze content of hot metal. This measurement operation gives information about the process and called inblow measurement. The phase from start of blowing oxygen till inblow measurement is called main blow. With this measurement, how much oxygen is going to blown is decided and dynamic blow phase starts. Dynamic blow ends when aims are achieved. At the end of blow sublance again moves downwards this time with Temperature Sample Oxygen (TSO) probe type. TSO probes measures oxygen amount and temperature of crude steel. With this sample it will be decided whether finishing the process or blowing more oxygen. Regarding analyze results, if steel sample content can not reach the desired content, reblow phase starts or again flux materials are added. If aims are achieved tapping starts, crude steel poured into the steel ladle, slag poured into slag pot and the process finished at BOF.

Operational steps described in upper paragraph can be seen schematically on Figure 1.6. A BOF production capacity is nearly 200 tons for one heat and working temperature is nearly 1700 deg. C. Graphic of oxygen amount consumption of a heat in BOF shown in detail on Figure 1.7. Also at the bottom of the graphic process, phases drawn with different color horizontal bars. Graphic explains when main blow, dynamic blow started and when chemical reactions like desiliconization and dephosporization started and finished.

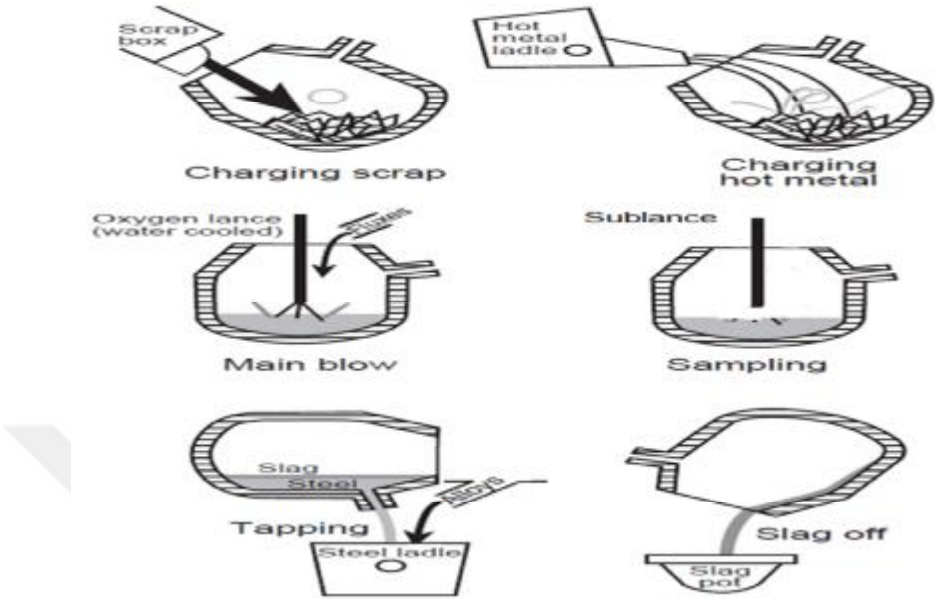

Figure 1.6. Schematic of operational steps in BOF (Miller et al.,1998)

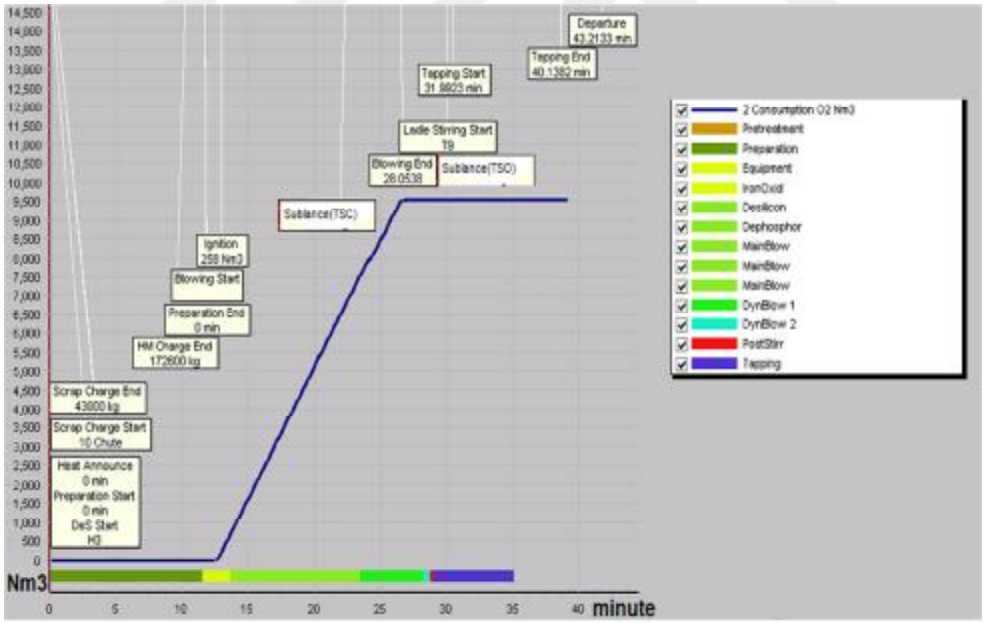

Figure 1.7. Graphic of oxygen blown to BOF and process phases of a heat

1.4. Chemical Reactions in BOF Steelmaking

Table 1.3. Chemical reactions occurring in a BOF (Brooks et al,2002)

Oxygen pick up by the metal	
$O_2(g) = 2\underline{O}$	(1)
$(FeO) = \underline{Fe} + \underline{O}$	(2)
$(Fe_2O_3) = 2(FeO) + \underline{O}$	(3)
$CO_2(g) = CO(g) + \underline{O}$	(4)
Oxidation of elements in the metal:	
$C + O = CO(g)$	(5)
$Fe + O = (FeO)$	(6)
$Si + 2O = (SiO_2)$	(7)
$Mn + O = (MnO)$	(8)
$2P + 5O = (P_2O_5)$	(9)
Oxidation of compounds in the slag	
$2(FeO) + ½O_2(g) = (Fe_2O_3)$	(10)
$2(FeO) + CO_2(g) = (Fe_2O_3) + CO$	(11)
Flux Reactions	
$MgO(s) = (MgO)$	(12)
$CaO(s) = (CaO)$	(13)
Gas Reactions	
$CO(g) + ½O_2(g) = CO_2$	(14)

Note:"_" metal phase and "()" slag phase

According to the study of Brooks & Coley (2002) in basic oxygen furnace, many chemical reactions can be observed. Detailed information can be found in Table 1.3 given above. Dissolution of oxygen into the metal from O2 gas which is numbered with 1; decarburization through dissolved oxygen which is numbered with 5 and oxidation of elements (Fe, Si, Mn, P) in the metal which are numbered

between 6 and 9 can be counted as the main reactions during blowing in basic oxygen furnace.

During blowing, solid and liquid oxides are the outcomes of the reactions and they form compounds with the lime added. As it can be seen from the reaction numbered with 5 from the Table 1.3 which shows gas formation of CO, droplets of liquid metal are transferred into slag which can be observed as foam. With other words, the slag in a BOF during oxygen blowing is an emulsion of liquid slag and metal droplets and foaming because of the influence of gas bubbles (Ruuska, 2012).

2. PREVIOUS WORKS

In the study of Min Han, Yang Li and Zhnaji Cao (2014), a model composed of Case-Based Reasoning (CBR) and Support Vector Machine (SVM) were utilized for calculating oxygen blowing volume. There are two stages for calculation of models in steel making process as main blow corresponds to static model and second blow to dynamic model. While in the main blow stage, the condition attributes of cases are hot metal temperature, hot metal carbon content, manganese content in hot metal, hot metal phosphorus content, silicon content, dolomite, lime, iron ball and scrap weight; in the second blow stage, the condition attributes of cases are the carbon content and temperature measured by sub-lance. The methods utilized in (Min Han et al, 2014) were tested by using 50 heats data of 180t BOF.

Precision of the models were evaluated by using two criteria which are Root Mean Square Error (RMSE) and Hit Rate (HR). Formulas of RMSE and Hit Rate that were used in (Min Han et al, 2014) were given in Eq. (2.1). As it can be seen from the formulas, different determinants are used for this calculation.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - r_i^2)}{n}} \qquad HR = \frac{N_{hit}}{N_{heats}} \tag{2.1}$$

p_i: prediction value

r_i: practical value

n: number of the calculated samples

N_{hit}: the number of the heats that hit the target. If the absolute value of the distance between the practical oxygen volume and the the calculated value is called as a hit.

N_{heats}: The number of calculated samples.

The results of the study can be seen from the Table 2.1. given below. (Min Han et al, 2014). It is reported that the threshold of Hit Rate (named as accuracy in the table) was chosen ±500 m3 for static model and ±300 m3 for dynamic model.

Table 2.1. Prediction model results of the oxygen blowing volume calculation (Min Han et al, 2014)

Prediction Model	Static		Dynamic	
	RMSE	Accuracy (±500m^3)	RMSE	Accuracy (±300m^3)
CBR	429.64	0.84	195.93	0.84
DE CBR	362.25	0.88	172.41	0.92
DE I-CBR	328.73	0.88	157.70	0.94

In their study Min Han and Yao Zhao (2011) used adaptive-network-based fuzzy inference system (ANFIS) algorithm to predict the volume of second-blow oxygen. Carbon content and temperature measured at the end of the main blow were chosen as inputs of the models. Training data consists of 300 data 100 heats production data. As in (Min Han et al, 2014), in this study hit rate and RMSE criteria were used to evaluate the performance of the model. Hit rate threshold was selected ±200 m^3 and hit ratio was found %86.

Table 2.2. Comparison of CBR and ANFIS of the study (Han and Zhao,2011)

Method	The volume of oxygen	
	Hit Ratio (%)	RMSE (m^3)
CBR	72	196.82
ANFIS	86	170.45

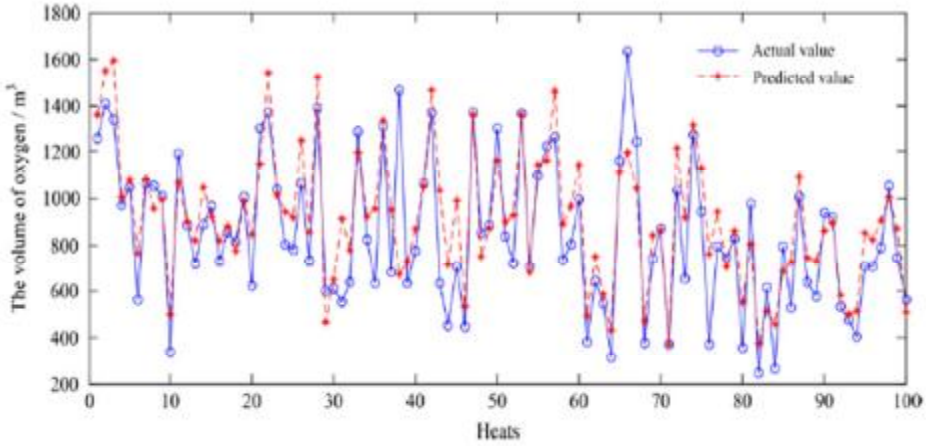

Figure 2.1. Prediction results of the (Han and Zhao,2011) study

Wang et al. (2012) applied combination of multiple linear regression and oxygen balance model to predict required oxygen amount. Independent variables used in this study were campaign life, charged scrap, total charge, hot metal temperature, charged hot metal, charged pig iron, hot metal ratio, silicon content in hot metal, manganese content in hot metal, phosphorus content in hot metal, sulphur content in hot metal. An integrated prediction model established based on multiplying weight coefficients with the result of regression equation and oxygen balance model equation of the required oxygen amount like the formula shown below.

$$V_{O_2} = w_r V_{regression,\ O_2} + w_b V_{balance,\ O_2} \tag{2.2}$$

268 heats used to determine the weight coefficients. For testing the success of the model 35 heats production data were used. Linear regression model, oxygen balance model and integrated model's relative error drawings can be seen on Figure 2.2 separately. When the relative errors were below 5%, the hit rate of the integrated model was 97.14% ; the hit rate of the model was 91.43% when the

relative errors were restricted within 2%. The average relative error of integrated model was 1%.

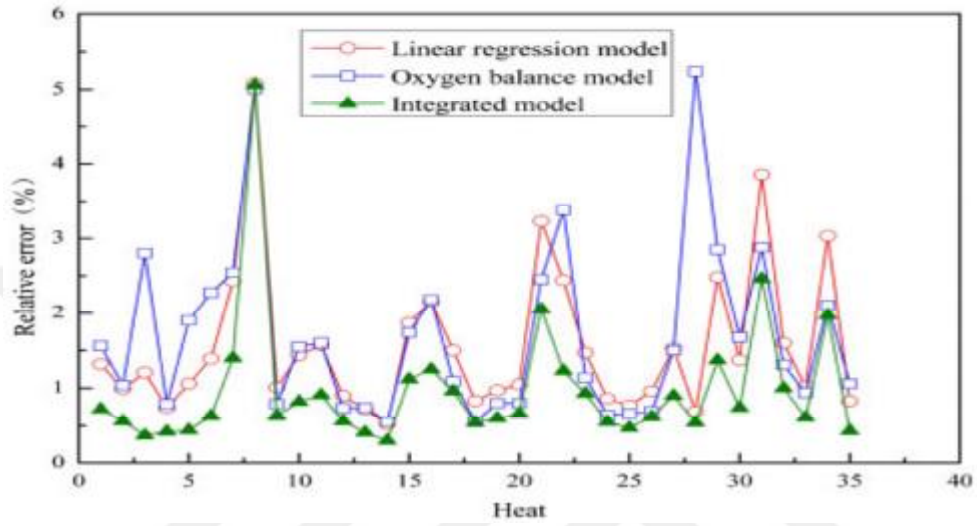

Figure 2.2. Relative errors of the models of the Wang et al (2012) study

Another study that predicted end blow oxygen amount is Cox et al. (2001). They used Artificial Neural Network (ANN) to predict how much oxygen to blow. The inputs of the study are aim carbon composition, aim temperature, sample carbon composition and sample temperature. To combine inputs, they subtract aim carbon composition from sample carbon composition and named it carbon removal required, and also subtract aim temperature from sample temperature and named it temperature rise required. By this way, the inputs reduced to two inputs, which are carbon removal, required and temperature rise required. They collect 1600 data from BOF and split into two even parts, one for training and one for test. Found results are; standard deviation of the end blow oxygen error: 225.31 m^3 and mean absolute error:179.75 m^3.

3. METHODS

3.1. Linear Regression

Let Y as response variable and $X_1, X_2, \ldots, X_K$ as explanatory variables then a general model with respect to these variables is

$$E[Y|X_1 = x_1, X_2 = x_2, \ldots, X_K = x_K] = \phi(x_1, x_2, \ldots, x_K) \tag{3.1}$$

However, to shorten brevity the expression, generally the conditioning part can be dropped and E[Y] is used

$$\phi(x_1, x_2, \ldots, x_K) = \beta_0 + \beta_1 x_1 + \cdots + \beta_K x_K \tag{3.2}$$

which with the parameter β_j, $\phi(x_1, x_2, \ldots, x_K)$ is linear. This limitation can be eliminated a proper transformation. For the gravitational model, use logarithm then get the straight line (Seber, 1977).

$$logF = loga - \beta logd. \tag{3.3}$$

For the linear model, the x_i might be functions of other variables z, w, etc.; for example, $x_1 = \sin z$, $x_2 = \log w$, and $x_3 = zw$. We may also have $x_i = x^i$, which leads to polynomial model; the linearity refers to the parameters, not the variables. Note that "categorical" models can be included under our umbrella by using indicator x-variables. For example, assume that we would like to analogize the

means of two populations, say, $\mu_i = E[U_i]$ $(i = 1, 2)$. Then we can blend the data into single model.

$$E[Y] = \mu_1 + (\mu_2 - \mu_1)x = \beta_0 + \beta_1 x \tag{3.4}$$

Where $x = 0$ when Y is a U_i observation and $x = 1$ when Y is a U_2 observation. Here $\mu_1 = \beta_0$ and $\mu_2 = \beta_0 + \beta_1$, the difference being β_1. We can extend this idea to the case of comparing m means using $m - 1$ dummy variables. In similar fashion, we can combine two straight lines,

$$U_j = a_j + \gamma_j (j = 1,2) \tag{3.5}$$

Using a dummy x_2 variable which takes the value 0 if the observation is from the first line, and 1 otherwise. The combines model is

$$\begin{aligned} E[Y] &= a_1 + \gamma_1 x_1 + (a_2 - a_1)x_2 + (\gamma_2 - \gamma_1)x_1 x_2 \\ &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \end{aligned} \tag{3.6}$$

say, where $x_3 = x_1 x_2$. Here $a_1 = \beta_0$, $a_2 = \beta_0 + \beta_2$, $\gamma_1 = \beta_1$, $\gamma_2 = \beta_1 + \beta_3$.

In the various models considered above, the explanatory variables may or may not be random. For example, dummy variables are nonrandom. With random X-variables, we perform the regression with the usual reserve on observed values, provided that they are measured exactly or at least with adequate accuracy.

3.2. Multi-Layer Perceptron

MLP as a type of ANN is composed of many artificial neurons that are linked together over multi-layers according to a specific network architecture. The

objective of the neural network is to transform the inputs into meaningful outputs (Kandel, 2000).

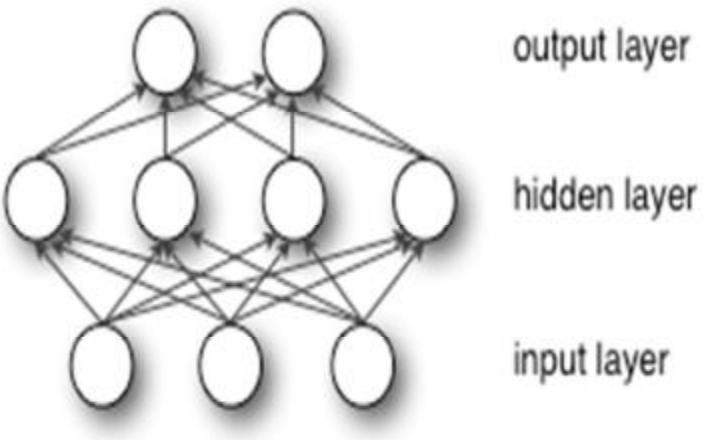

Figure 3.1. Representation of MLP, (Kandel, 2000)

This is all to say, all nodes of each layer are totally linked to all nodes of the next layer. Since Artificial Neural Network has the ability to learn, training diagram of multi-layer perceptron is based on backpropagation approach, which gets the error rate by calculating difference of real outputs and desired outputs. Then the error is implemented to updated function of network weights in order to establish best fitted network to the main pattern of training data. The square error rate between the real output and the desired output of p^{th} input pattern is shown as follows;

$$E_P = \frac{1}{2}(d^p - y^p)^2 = \frac{1}{2}\sum_{j=1}^{S}(d_j^p - y_j^p)^2 \tag{3.7}$$

Where d^p and y^p are the desired output vector and the real output vector, respectively, d_j^p and y_j^p are the desired and the real outputs for j^{th} node of output

layer, respectively s is the size of the output vector. Consequently, the total error rate for the p patterns is computed as;

$$E = \sum_{p=1}^{p} Ep = \frac{1}{2} \sum_{p=1}^{p} \sum_{j=1}^{S} (d_j^p - y_j^p)^2 \tag{3.8}$$

Where P is the total number of input pattern, s is the total number nodes in the output layer, d_j^p is the desired output of j^{th} node of output layer, and y_j^p is real output of p^{th} input pattern produced by j^{th} node. In multi-layer perceptron, each connection point between adjacent layers have a weight that should be updated iteratively to decrease the error rate E to acceptable. The updating formula is

$$w_{ij}(\mathrm{t}+1) = w_{ij}(t) + \eta \times \Delta w_{ij}(t) + a \times w_{ij}(t-1)$$
$$\Delta w_{ij}(t) = -\left(\frac{\partial E_p}{\partial w_{ij}(t)}\right) \tag{3.9}$$

Where η is learning rate, a moment coefficient, $w_{ij}(t)$ is weight of connection between the node i and j at time t and $w_{ij}(t+1)$ is the weight of that connection next time. The learning operation of multi-layer perceptron will be completed when either error rate reach to an acceptable value or reach to maximum number of iteration defined earlier.

3.3. Radial Basis Function Network

Radial functions are simply class functions and they could be used a model as either linear or nonlinear. Also considering network, they could be either single layer or multi-layer network (Orr, 1996).

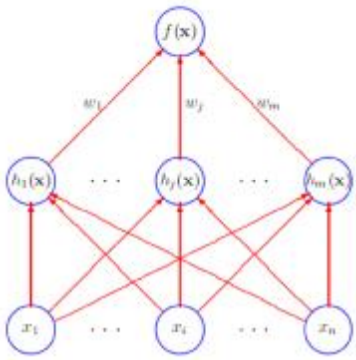

Figure 3.2. Representation of RBF

The classic radial basis function network can be seen on the figure above. Each of *n* components of the input vector x feeds forwardly to *m* basis function. Then their outputs are linearly updated with weights into the network output *f*(x).

When the basis functions can move or change the size or have more than one hidden layer, RBF network become nonlinear. Here is avoided using expensive nonlinear gradient descent algorithms such as the conjugate gradient and variable metric methods (Teukolsky, 1996) that are employed explicitly nonlinear networks.

The most common formula for radial basis function is

$$h(\mathrm{x}) = \phi((\mathrm{x}-\mathrm{c})^{T}\mathrm{R}^{-1}(\mathrm{x}-\mathrm{c})) \tag{3.10}$$

ϕ is the function used, c shows the center and R denotes the metric and $(\mathrm{x}-\mathrm{c})^{\mathrm{T}}\mathrm{R}^{-1}(\mathrm{x}-\mathrm{c})$ gives the distance between x and the center c in the metric define by R. Here $\phi(z)$ function may be Gaussian, multi-quadratic, inverse multi-quadratic. Generally, the metric is Euclidean. Hence $\mathrm{R} = r^{2}\mathrm{I}$ for some scalar radius *r* and the Equation simplifies to

$$h(\mathrm{x}) = \phi((\mathrm{x}-\mathrm{c})^T(\mathrm{x}-\mathrm{c})/r^2) \tag{3.11}$$

More simplificated one is one-dimensional input space in which case

$$h(\mathrm{x}) = \phi((\mathrm{x}-\mathrm{c})^2/r^2). \tag{3.12}$$

3.4. Extreme Learning Machine

Assuming Single Layer Feed Forward Networks (SLFNs) with L hidden nodes may be represented by the following equation (Tang, 2016):

$$\mathrm{f_L}(\mathrm{x}) = \sum_{\mathrm{i}=1}^{\mathrm{L}} \mathrm{G_i}(\mathrm{x}, \mathrm{a_i}, \mathrm{b_i}) \,.\, \beta_\mathrm{i},\ \mathrm{a_i} \in \mathrm{R^d},\ \mathrm{b_i}, \beta_\mathrm{i} \in \mathrm{R} \tag{3.13}$$

where $Gi(\cdot)$ shows the i^{th} hidden node activation function, $\boldsymbol{a}_i$ is the input weight vector connecting the input layer to the i th hidden layer, b_i is the bias weight of the i th hidden layer, and β_i is the output weight. For additive nodes with activation function g , G_i is defined as follows:

$$\mathrm{G_i}(\mathrm{x}, \mathrm{a_i}, \mathrm{b_i}) = \mathrm{g}(\mathrm{a_i} \,.\, \mathrm{x} + \mathrm{b_i}) \tag{3.14}$$

and for RBF nodes with activation function g , G_i is defined as

$$G_i = g\left(\frac{\|x - a_i\|}{b_i}\right),\ a_i \in R^d,\ b_i \in R^+. \tag{3.15}$$

(Huang et al, 2006) have proved that the SLFNs have ability to approximate any continuous target functions over any compact subset $X \in R^d$ with above arbitrary initialized customizable or RBF nodes. Let $L^2(X)$ be a space of functions f on a compact subset X in the d -dimensional Euclidean space R^d such that $|f|^2$ is

integrable, that is, $\int_X |f(x)|^2 dx < \infty$. For $u,v \in L^2(X)$, the inner product $\langle u,v \rangle$ is defined by

$$< u, v > = \int_X u(x)v(x)dx. \quad (3.16)$$

The norm in $L^2(X)$ space is typified as $\|\cdot\|$, and the closeness between network function f_n and the target function f is measured by the $L^2(X)$ distance

$$\| f_L - f \| = \left(\int_X |f_n(x) - f(x)|^2 dx\right)^2 . \quad (3.17)$$

Theorem **:**

Given any bounded unconstant in pieces continuous function g : $\mathbf{R} \rightarrow \mathbf{R}$, if span $\{G(\mathbf{a},b,\mathbf{x}) : (\mathbf{a},b) \in \boldsymbol{R}^d \times R\}$ is intense in L^2 , for any target function f and any function sequence $g_L(x)=G(\mathbf{a}_L,b_L,\mathbf{x})$ arbitrarily generated according to any continuous sampling distribution, $\lim_{n\rightarrow\infty}\|f-f_n\| = 0$ holds with probability one if the output weights βi are determined by common least square to minimize $\|f(x)-\sum_{i=1}^{L}\beta_i g_i(x)\|$.

The theorem above (Huang, 2006), (Huang, 2008), (Huang, 2012) shows that arbitrarily produced networks with the outputs being solved by least mean square are able to pursue the approximation capability, if and only if the activation function g is unconstant piecewise and span $\{ G(\mathbf{a},b,\mathbf{x}) : (\mathbf{a},b) \in \boldsymbol{R}^d \times R \}$ is dense in L^2.

According to the Theorem, the ELM can be built with arbitrarily initialized hidden nodes. Given a training set $\{(\mathbf{x}_i , \mathbf{t}_i)| \mathbf{x}_i \in \mathbf{R}^d , \mathbf{t}_i \in \mathbf{R}^m, i = 1, \ldots , N\}$, where $\mathbf{x}_i$ is the training data vector, $\mathbf{t}_i$ typifies the target of each sample, and L denotes the number of hidden nodes. Dissimilar traditional learning algorithms, ELM theory

purpose is to reach the smallest error but also the smallest norm of output weights (Huang, 2014).

$$\text{Minimize: } \| \beta \|_u^{\sigma_1} + \lambda \| \mathbf{H}\beta - \mathbf{T} \|_v^{\sigma_2} \tag{3.18}$$

where $\sigma_1 > 0$, $\sigma_2 > 0$, u, v = 0, (1/2), 1, 2, . . . ,+ ∞, **H** is the hidden layer output matrix

$$\mathbf{H} = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} \tag{3.19}$$

and **T** is the training data target matrix

$$\mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix} = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \ddots & \vdots \\ t_{N1} & \cdots & t_{Nm} \end{bmatrix} \tag{3.20}$$

The ELM training algorithm can be summarized as follows (Huang,2012).

1) Randomly assign the hidden node parameters, e.g., the input weights a_i and biases b_i for additive hidden nodes, $i = 1, \ldots, L$.

2) Calculate the hidden layer output matrix **H**.

3) Obtain the output weight vector.

$$\boldsymbol{\beta} = \mathbf{H}\dagger\mathbf{T} \tag{3.21}$$

where $\mathbf{T} = [\mathbf{t}1, \ldots, \mathbf{t}_N]^T$,$\mathbf{H}\dagger$ is the Moore–Penrose generalized inverse of matrix **H**.

The orthogonal projection method can be efficiently used for the calculation of MP inverse: $\mathbf{H}\dagger = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$, if $\mathbf{H}^T\mathbf{H}$ is nonsingular; or $\mathbf{H}\dagger = \mathbf{H}^T(\mathbf{H}^T\mathbf{H})^{-1}$, if $\mathbf{H}\mathbf{H}^T$ is nonsingular. According to the ridge regression theory, it was suggested that a positive value $(1/\lambda)$ is added to the diagonal of $\mathbf{H}^T\mathbf{H}$ or $\mathbf{H}\mathbf{H}^T$ in the calculation of the output weights β. By doing so, according to (Teukolsky, 1996) and (Huang, 2014), the resultant solution is equivalent to the ELM optimization solution with $\sigma 1 = \sigma 2 = u = v = 2$, which is more steady and has better generalization performance. That is, in order to enhance the stability of ELM, we can have

$$\beta = \mathbf{H}^T\left(\frac{1}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \tag{3.22}$$

and the corresponding output function of ELM is

$$\begin{aligned} f(x) &= h(x)\beta = h(x)\mathbf{H}^T\left(\frac{1}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \\ f(x) &= h(x)\beta = h(x)\left(\frac{1}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{H}^T\mathbf{T}. \end{aligned} \tag{3.23}$$

3.5. K-Nearest Neighbor Algorithm

KNN is based on the idea of using training data as the basis for predicting test cases or future values. “Neighbors” are selected from training data, and concatenated according to certain distance measures between the attributes of the training data and the attributes of the testing data. Output will be predicted by calculating the average of target values of the neighbors. Let D_i as the distance between i^{th} training data and the testing data, , $x_{training}$ and $x_{testing}$ as the attribute

values of training data and test data respectively, calculation is defined as follows (Liu, 2016).

$$D_i = \sum_{j=1}^{n} w_j \mid x_{training,i,j} - x_{testing,j} \mid \tag{3.25}$$

In Eq. (3.25), i and j denote the indices of training data and of the attributes, n is the total number of attributes, and w_j is the weight of j^{th} attribute. The values of w_j are determined by utilizing k-fold Cross Validation. The target values of K nearest neighbors are used to generate the prediction value. Formula is given in Eq. (3.26) where P denotes the predicted value, $T_{training}$ is the target value of training data, i represents the index of chosen neighbors, and K is the total number of nearest neighbors. v_i is the weight of ith target value.

$$P = \frac{\sum_{i=1}^{K} v_i T_{training,i}}{\sum_{i=1}^{K} v_i} \tag{3.26}$$

In the literature, there exist three modified KNN algorithms based on the main KNN concept, which are named as Distance Weighted KNN, KNN with Gradient Descent, and Local Weighted Multivariate Linear Regression model.

In Distance Weighted KNN model, the weights of target values (v_i) are determined by the distance (D_i) between the training data and the testing data, which is calculated as $v_i = 1/D_i^2$, where i is the index of selected neighbors from training data.

In KNN with Gradient Descent model, the weights of target values are trained by the Gradient Descent (GD) algorithm. Gradient Descent is a first order optimization algorithm that is utilized to find the local minimum of a quadratic function. It uses the idea of taking steps proportional to the negative gradient of the objective function. In ANN, the GD method is generally used for minimizing the

squared error by updating the weights. Conventionally, the squared error is defined as in Eq. (3.27)

$$E = \frac{1}{2} \sum_{d \in TS} (T_{testing,d} - T_{prediction,d})^2 \quad (3.27)$$

where E represents squared error, $T_{testing}$ denotes the target value of testing data and $T_{prediction}$ is the predicted target value. Weights are updated in accordance with the rule given in Eq. (3.28), where the v_i are the weights of training data target values, i is the index of neighbors, Δv_i is the step change at each iteration, and $T_{testing}$, $T_{training}$,and $T_{prediction}$ are the target values of testing data and training data, and predicted value, respectively. δ is the step size coefficient.

$$\Delta v_i = \Delta v_i + \delta \left(T_{testing} - T_{prediction}\right) T_{training,i} \quad (3.28)$$
$$v_i = v_i + \Delta v_i$$

In KNN with Gradient Descent model, GD process described above is used together with k-fold Cross Validation to calculate the weights.

Local Weighted Multivariate Linear Regression model is a hybrid algorithm, which implements combination of Multivariate Linear Regression (MLR) algorithm and KNN algorithm. Unlike standard KNN, for each testing data an MLR model is generated with its neighbors. The MLR can be described as in Eq. (3.29)

$$T = \sum_{j=1}^{n} \beta_j x_j + \varepsilon \quad (3.29)$$

where T, x, β are the target value, the attribute, the corresponding coefficient, respectively and ε represents the disturbance term or noise term. In this model, firstly β and ε are estimated by training MLR on neighbors, and then test result is

predicted by using the attributes of testing data. This hybrid model overcomes the nonlinearity limitation of the linear regression model.

3.6. Support Vector Machine

SVM is introduced for regression and classification problems by Vapnik. SVM regression is based on the idea of predicting the value of the given function within a predefined error threshold ϵ. Let $\{x_i,y_i\}^{N}_{i=1}$ as dataset where y_i is the target output for training data x_i, and N denotes the number of data (Kaushik, 2016).

A line approximation by using training datasets can be considered as $y'=w\varphi(x)+b$ where w is the weight matrix, $\varphi(x)$ is the mapping function and b is the bias. An error margin ϵ is defined such that for any predicted data y'_i,($y_i - y'_i$) $\leq \pm \epsilon$ must be satisfied. The error function is given in Eq.(3.30)

$$\frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N}(\xi_i^+ + \xi_i^-) \tag{3.30}$$

where C is the penalty parameter, ε_i^+ and ε_i^- are the slack variable penalties when the training sample lies outside and above or below the epsilon insensitive tube, respectively. Error function is minimized by using a set of Lagrange multipliers α^+ and α^- obtained for each training data with equations for w and b given as

$$w = \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)\varphi(x_i)$$
$$b = \frac{1}{N}\sum_{i=1}^{N}(y_i - \varepsilon - \sum_{m=1}^{N}(\alpha_i^+ - \alpha_i^-)K(x_i, x_m)) \tag{3.31}$$

The SVM regression/prediction equation is given in Eq.(3.32), where x'is the input test data, $K(x_i,x') = < \varphi(x_i)'\varphi(X'$» is the Kernel function and y' is the predicted output.

$$y' = \sum_{m=1}^{N}(\alpha_i^+ - \alpha_i^-)K(x_i, x') + b \quad (3.32)$$

The value ($\alpha_i^+ - \alpha_i^-$) is non-zero for all support vectors and all the support vectors are employed to build the regression model. An attempt is made to prune the number of support vectors required by the SVM model, using an optimum threshold technique that employs the support vectors satisfying the condition ($\alpha_i^+ - \alpha_i^-$) $\leq \alpha_{threshold}$. Optimum threshold is the $\alpha_{threshold}$ value for which aught or negligible degradation is observed.

$$k(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0 \quad (3.33)$$

The formula of the RBF kernel used in SVM is given above (Hofmann, 2008). When training SVM with the RBF kernel, two regulating parameters must be considered: C and γ. The parameter γ plays very important role for the performance of the kernel, and should be carefully tuned to the problem. If γ is overestimated, the kernel exponential function will act as linear and the feature space will lose its non-linearity. If γ is underestimated, the exponential function will lack regularization.

The penalty parameter C, popular in all SVM kernels, controls trade-offs between the margin and the size of the slack variables. A small C will lead to higher training errors, whereas a large C will lead to a performance similar to that of a hard-margin SVM.

Also, there is an optimization method for training SVMs that SMO algorithm. SMO does not need a quadratic solver. With this feature SMO is different from the other SVM algorithms. SMO finds two Lagrange multipliers continuously which can be optimized with respect to each other and calculates ideal steps for the two Langrange multipliers. When SMO could not find optimizable two Lagrange multipliers, the original QP problem is solved. SMO

actually is formed of two parts: (1) a set of heuristics for selecting pairs of Lagrange multipliers adequately to work on, and (2) size two QP problem's analytical solution (Flake et al, 2002)

.

4. RESULTS AND DISCUSSIONS

BOF steelmaking is a highly complex multiphase physics and chemical reaction process, which aims to reduce the carbon content and increase temperature. The whole process does not require external heat source and it comes from the exothermic reaction of oxygen and molten iron of carbon, silicon and other elements (Min H. et al,2014)

Table 4.1. Description of variables used in BOF steelmaking

Variables	Unit	Mean	Standard Deviation
Aim Temperature	deg C	1660	16.243
Aim Carbon Content	wt%	0.04	0.008
TSC Temperature	deg C	1555	19.841
TSC Carbon Content	wt%	0.638	0.158
Hot Metal Weight	ton	177.872	4.241
Scrap Weight	ton	37.122	2.653
Bath Level	mm	2318.977	242.928
Dynamic Blow O2 Amount	Nm^3	2069.824	296.827

Data collected from 200 tons BOF steel production in this study. 3988 sets of data used for prediction of oxygen amount used at dynamic blow stage. There are 7 variables chosen as inputs which are aim temperature, aim carbon content, TSC temperature, TSC carbon content, hot metal weight, scrap weight and bath level and one output dynamic blow O2 content.

Weka is used for running the machine learning algorithms to predict the oxygen amount. The Weka workbench is a collection of state-of-the-art machine learning algorithms and data preprocessing tools. It is designed so that one can quickly try out existing methods on new datasets in flexible ways. It provides extensive support for the whole process of experimental data mining, including

preparing the input data, evaluating learning schemes statistically, and visualizing the input data and the result of learning. As well as a wide variety of learning algorithms, it includes a wide range of preprocessing tools.

Weka uses 5 evaluation parameters in its default settings. These are Correlation Coefficient (CC), Root Mean Square Error (RMSE), Root Relative Square Error (RRSE), Mean Absolute Error (MAE) and Relative Absolute Error (RAE). These evaluation parameters formulas are given below:

$$CC = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{4.1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{4.2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4.3}$$

$$RAE = \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}(\overline{y} - y_i)^2} \tag{4.4}$$

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}(\overline{y} - y_i)^2}} \tag{4.5}$$

x_i: Prediction value

y_i: Practical value

$\overline{x}$: Mean of prediction values

$\overline{y}$: Mean of practical values

n: Number of calculated values

10-fold cross validation is used to estimate how accurately models will perform in practice. Grid Search is used for finding optimal parameter for a prediction model. Grid Search is a meta-classifier in Weka for exploring two parameters, hence the grid in the name. The initial grid is worked on with 2-fold cross validation to determine the values of the parameter pairs for the selected type of evaluation. The best point in the grid is then taken and a 10-fold cross validation is performed with the adjacent parameter pairs. If a better pair is found, then this will act as new center and another 10-fold CV will be performed. This process is repeated until no better pair is found or the best pair is on the border of the grid.

Table 4.2. Weka results

Evaluation Parameters	**Prediction Models**				
	SMO Reg	**MLP**	**RBF Regression**	**Linear Regression**	**IBk**
CC	0.785	0.784	0.777	0.772	0.771
RMSE	183.875	183.662	186.278	188.096	192.726
MAE	141.771	142.599	144.914	146.406	150.244
RAE	0.594	0.597	0.607	0.613	0.629
RRSE	0.620	0.620	0.628	0.635	0.650

Based on the Weka results as seen on Table 4.2 all prediction models are very close to each other. Best prediction model is SMO regression model. RBF kernel used as a kernel and the best values of parameters C, γ is chosen with grid

search. Best values are searched within the intervals $[10^{-3}, 10^{3}]$ for C parameter and $[10^{-3}, 10^{2}]$ for γ parameter. The best performance with CC=0.785, RMSE=183.875, MAE=141.771, RAE=0.594, RRSE=0.620 evaluation parameter values are found with C=15, $\gamma = 0.13$.

MLP's evaluation parameters are nearly same with SMO regression prediction model. Again with grid search method best performance searched for learning rate and momentum parameters both with the interval $[10^{-5}, 1]$. The best performance with CC=0.784, RMSE=183.662, MAE=142.599, RAE=0.597, RRSE=0.620 evaluation parameter values are found with learning rate=0.01 and momentum=0.001 and one hidden layer with 7 neurons.

Even though the correlation coefficients of RBF, linear regression and IBk prediction models are equal, RBF regression model is slightly better in RMSE and MAE values. For RBF, scale optimization option determines whether one global sigma value is used for all units (fastest), whether one value is used per unit , or a different value is learned for every unit/attribute combination. Best result CC=0.777, RMSE=186.278, MAE=144.914, RAE=0.607, RRSE=0.628 received with the scale per unit option.

IBk is the name for k-nearest neighbors algorithm in Weka. IB is abbreviation for Instance Based. KNN parameter changed between [1, 200] for finding the best results. The best result is found with using KNN=47 and Euclidean distance.

After receiving these results by working with Weka, algorithms are also run with Matlab with same dataset. As shown on Table 4.3 that ELM is also added to algorithms and it gave nearly the same results with SMO regression. It can be seen that there is not much more difference in the evaluation parameters when the Table 4.2 Weka results and Table 4.3 Matlab results are compared.

Table 4.3. Matlab results

Evaluation Parameters	**Prediction Models**				
	ELM	**SMO Reg.**	**Linear Regression**	**RBF Regression**	**MLP**
CC	0.787	0.787	0.775	0.774	0.747
RMSE	188.659	189.002	193.381	201.554	203.657
MAE	146.826	146.924	150.732	157.876	158.951
RAE	0.626	0.627	0.635	0.673	0.678
RRSE	0.649	0.650	0.656	0.692	0.700

5. CONCLUSION

The most important stage during the conversion of iron to steel in BOF steelmaking process is oxygen-blowing stage. Amount of oxygen blown into BOF has a great importance for the refractory life of BOF, production cost and producing steel in desired quality.

In this study, different algorithms, SMO Regression, MLP, RBF Regression, ELM, IBk and linear regression is used for develop prediction models of oxygen amount blown into BOF at dynamic blow stage. Weka and Matlab is used for running the prediction models. So that the accuracy of the produced results is verified by comparing the results of these two instruments. Since there is no need to write code, it is concluded that Weka is one-step ahead of Matlab in terms of usage.

Grid Search is used for finding optimal parameter for these prediction models. When evaluation parameters CC, RMSE, RRSE, MAE and RAE are considered the results of the models are close to each other. ELM and SVM based SMO predicted slightly better results than the other models with gathered production data from BOF.

For future work, other factors affecting the process can be take into account and with considering different inputs; use of feature selection methods may become compulsory. Although the most common machine learning algorithms are used in this study, popular approach in recent years “combining algorithms” can also help to predict the output more accurate.

REFERENCES

World Steel in Figures, 2016. World Steel Association. ISBN 978-2-930069-87-6, p7.

Global Steel Report, 2016. USA Department of Commerce International Trade Administration, p5.

Turkish Steel Industry, 2015. Turkish Steel Producers Association, p1.

Steel Statistical Yearbook, 2016. World Steel Association.

Chen Peng, Lu Yong-zai,2010. Memetic algorithms-based neural network learning for basic oxygen furnace endpoint prediction. Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering), p841-848.

Miller T.W., Jimenez J., Sharan A., Goldstein D.A., 1998. The Making, Shaping and Treating of Steel, 11th Edition Steelmaking and Refining Volume. The AISE Steel Foundation, Pittsburgh, p475.

Brooks G.A. & Coley K.S.,2002 Interfacial area in top blown oxygen steelmaking. 61stIronmaking Conference Proceedings, Nashville, USA, p837–850.

Ruuska Jari,2012. Special measurements and control models for a Basic Oxygen Furnace (BOF). University of Oulu, p20.

Min H., Yang L., Zhanji C.,2014. Hybrid intelligent control of BOF oxygen volume and coolant addition. Neurocomputing 123, p415–423

Min H., Zhao Y., 2011. Dynamic control model of BOF steelmaking process based on ANFIS and robust relevance vector machine, Expert Systems with Applications 38, p14786–14798

Wang Z., Liu Q., Xie F. M., Wang B., Wang B., Lu X. C. & Fu G. Q.,2012. Model for prediction of oxygen required in BOF steelmaking, Ironmaking & Steelmaking 39:3, p228-233.

Cox I.J., Lewis R.W., Ransing R.S., Laszczewski H., Berni G.,2001. Application of neural computing in basic oxygen steelmaking, Journal of Materials Processing Technology 120, p310–315

Min H., Zhanji C., Yang L., 2014. An Improved Case-Based Reasoning Method Based On Fuzzy Clustering and Mutual Information, Fifth ınternational Conference on Intelligent Control and ınformation Processing, p293-300

Huang G.B., Chen L., 2007. Convex incremental extreme learning machine. Neurocomputing, vol. 70, no. 16, p3056-3062.

Huang G.B., Chen L., 2008, Enhanced random search based incremental extreme learning machine. Neurocomputing, vol. 71, no. 16, p3460-3468.

Huang G.B., Zhou H., Ding X., and Zhang R., 2012, Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 2, p513–529.

Huang G.B., 2014, An insight into extreme learning machines: Random neurons, random features and kernels. Cognit. Comput., vol. 6, no. 3, p376–390.

Hofmann T., Schlkopf B., and Smola A.J., 2008. Kernel methods in machine learning. The annals of statistics p1171 – 1220

Huang G.B., Chen L., and Siew C.K., 2006. Universal approximation using incremental constructive feedforward networks with random hidden nodes *IEEE Trans. Neural Netw.*, vol. 17, no. 4, p879-892.

Tang J., Deng C. and Huang G.B., 2016. Extreme Learning Machine for Multilayer Perceptron. IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 4

Orr M.J.L., 1996. Introduction to Radial Basis Function Networks Centre for Cognitive Science University of Edinburgh, p5-25

Liu Z., Zhang Z., 2016. Solar Forecasting by K-Nearest Neighbors Method with Weather Classification and Physical Model, 978-1-5090-3270-9/16/$31.00 © IEEE

Kaushik K. and Manikandan J., 2016. SAR Image Compression using Optimum Threshold based Support Vector Machines. 978-1-5090-3646-2/16/$31.00 © IEEE

Kandel E. R., Schwartz J. H. and Jessell T. M. , 2000. Principles of neural science vol. 4, McGraw-Hill, New York

Seber G.A.F., Lee A. J., 1977. Linear Regression Analysis Second Edition, p4

Press W.H., Teukolsky S.A., Vetterling W.T., and Flannery B.P., 1992. Numeric Recipes in C Cambridge University Press Cambridge UK second edition

Flake G.W., Lawrence S., 2002. Efficient SVM Regression Training with SMO, Machine Learning, 46, p271–290.

CURRICULUM VITAE

Soner TÜRKOĞLU was born in Adana in 1983. He completed Adana Anatolian High School in 2001. He graduated from Fırat University Electrical and Electronic Engineering in 2005. He has work experiences as a Level 1 and Level 2 Automation Engineer. He likes playing basketball and reading books.