ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Fahriye GEMCİ

DETECTION OF REMOTE HOMOLOGY IN PROTEINS BY MACHINE LEARNING ALGORITHMS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2022

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

DETECTION OF REMOTE HOMOLOGY IN PROTEINS BY MACHINE LEARNING ALGORITHMS

Fahriye GEMCİ

PhD THESIS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of the degree of the Electrical and Electronics Engineering of Doctorate by the board of jury on 09/09/2022.

……	……	……
Prof. Dr. Ulus ÇEVİK	Prof. Dr. Turgay İBRİKÇİ	Prof. Dr. Zeynel CEBECİ
SUPERVISOR	II. SUPERVISOR	MEMBER
……..	……..	……..
Assoc.Prof.Dr.Sami ARICA	Assoc.Prof.Dr.Lütfü SARIBULUT	Asst.Prof.Dr.ErkutTEKELİ
MEMBER	MEMBER	MEMBER

PhD Thesis is written at the Department of Biotechnology of Institute of Natural and Applied Sciences of Çukurova University
Registration Number:

Prof. Dr. Sadık DİNÇER
Director
Institute of Natural and Applied Sciences

This study supported by Çukurova University Scientific Research Projects Unit.
Project No: FYL-2019-11621

ABTRACT

PhD THESIS

DETECTION OF REMOTE HOMOLOGY IN PROTEINS BY MACHINE LEARNING ALGORITHMS

Fahriye GEMCİ

Supervisor : Prof. Dr. Ulus ÇEVİK
II. Supervisor : Prof. Dr. Turgay İBRİKÇİ
Year 2022, 112 pages
Juri : Prof. Dr. Ulus ÇEVİK
Assoc. Prof. Dr. Sami ARICA
Assoc. Prof. Dr. Lütfü SARIBULUT
Prof. Dr. Zeynel CEBECİ
Asst. Prof. Dr. Erkut TEKELİ

The subject of this thesis is to develop a machine learning algorithm application that accurately performs remote homologous protein detection, which is an important problem in the field of bioinformatics.

The discovery of remote homolog proteins is important because it is beneficial to discover the structure of unknown proteins. In the thesis, the problem of different lengths of protein sequences is solved by using natural language processing methods such as the bag of words model. The performances were measured by applying motifs of different lengths as protein features. A new application in this thesis provides a solution to the unbalanced data problem. This application, which is a KNN method with k-split with various distance methods, is a competitive study.

Remote homologous proteins are a difficult problem to solve because they rely on small sequence similarities. In the thesis, another new application that trains with a new deep neural network that balances TF-IDF feature vectors calculated over n-grams with smoothing operations is carried out. The new application demonstrates the power of deep learning algorithms. The new application achieves better performance and overcomes the unbalanced data set.

Keywords: Bioinformatics, Remote Homologous Protein Detection, Deep Learning

ÖZET

DOKTORA TEZİ

UZAK HOMOLOG PROTEİNLERİN MAKİNE ÖĞRENME ALGORİTMALARI KULLANILARAK TESPİTİ

Fahriye GEMCİ

Danışman : Prof. Dr. Ulus ÇEVİK
II. Danışman : Prof. Dr. Turgay İBRİKÇİ
Yıl 2022, 112 sayfa
Jüri : Prof. Dr. Ulus ÇEVİK
Doç. Dr. Sami ARICA
Doç. Dr. Lütfü SARIBULUT
Prof. Dr. Zeynel CEBECİ
Dr. Öğr. Üyesi Erkut TEKELİ

Bu tezin konusu, biyoinformatik alanında önemli bir problem olan uzaktan homolog protein tespitini doğru bir şekilde gerçekleştiren bir makine öğrenmesi algoritması uygulaması geliştirmektir.

Uzak homolog proteinlerin keşfi, yapısı bilinmeyen proteinleri keşfetmekte faydalı olduğu için önemlidir. Bu tezde, farklı uzunluktaki protein dizileri problemi, kelime çantası modeli gibi doğal dil işleme yöntemleri kullanılarak çözülmüştür. Bu tez çalışmasının performansları, protein özellikleri olarak farklı uzunluklarda motifler uygulanarak ölçülmüştür. Bu tezde yeni bir uygulama, dengesiz veri sorununa çözüm sunmaktadır. Çeşitli uzaklık yöntemleri ile k-split ile bir KNN yöntemi olan bu yeni uygulama, diğer çalışmalarla rekabet edebilecek niteliktedir.

Uzak homolog proteinler, küçük dizi benzerliklerine dayandıkları için çözülmesi zor bir problemdir. Tezde, n-gram üzerinden hesaplanan TF-IDF öznitelik vektörlerini yumuşatma işlemleri ile dengeleyen yeni bir derin sinir ağı ile eğiten yeni bir uygulama daha gerçekleştirilmiştir. Bu yeni uygulama, derin öğrenme algoritmalarının gücünü göstermektedir. Bu yeni uygulama iyi bir performans ile dengesiz veri seti probleminin üstesinden gelmektedir.

Anahtar Kelimeler: Biyoinformatik, Uzak Homolog Protein Tespiti, Derin Öğrenme

GENİŞLETİLMİŞ ÖZET

Günümüzde hızla artan biyolojik veri, bu verilerden anlamlı bilgi çıkarılması ihtiyacını doğurmaktadır. Bilgisayarlar büyük miktarda veriyi işleyip analiz edebilme yeteneğine sahiptir. Bilgisayarların bu özelliğinden dolayı, biyolojik veriler bilgisayar algoritmaları ile işlenip anlam çıkarılmaktadır. Bundan dolayı; Molekuler Biyoloji ve Bilgisayar bilimini birleştiren bioinformatik alanı doğmuştur.

Homolog ve uzak homolog protein tespiti; biyoinformatik alanındaki çözülmesi gereken önemli problemlemlerden ikisidir. Proteinlerin homolojisini ve uzak homolojisini tanımak, yapısı ve işlevi bilinmeyen proteinler hakkında bilgi edinilmesine yardımcı olacağı için önemlidir. Yapısı bilinmeyen proteinler hakkında bilgi edinmek, tıpta hastalık tanısı ve yeni ilaç keşfinde çok faydalı olmaktadır.

Bu tezde çözülmesi homolog protein tespitine göre daha zor olan uzak homolog protein tespiti çalışması gerçekleştirildi. Uzak homolog proteinler, homolog proteinlere oranla daha düşük benzerliğe sahip oldukları için tespiti daha zor bir problemdir.

Bu tezde uzak homolog protein tespiti için, proteinleri yapılarına ve fonksiyonel özelliklerine göre sınıflandıran SCOP veritabanı kullanılmaktadır. SCOP veritabanından alınan proteinler, protein sıra bilgisine ilave olarak protein aile, süperaile ve fold bilgileri ile birlikte çekilir. Günümüzde sürekli yeni proteinler keşfedilmeye başlandığı için proteinleri düzenli olarak depolayacak veritabanı ihtiyacı artmaktadır. Bu sebeple, biyoinformatik problemlerinde büyük veri çözümlerine ihtiyaç olduğu gözlenmiştir. Artan protein verisini depolayabilmek ve gerektiğinde tekrar erişebilmek amacıyla çalışmada NoSQL veritabanında protein depolaması da gerçekleştirilmiştir.

Homolog proteinler aynı aileden olan proteinler iken; uzak homolog proteinler, aynı süperaileden olup farklı aileden olan proteinler olarak tanımlanırlar. Bu bilgiden yola çıkılarak proteinler hem uzak homolog ve hem uzak homolog

olmayan proteinler olmak üzere bu çalışmada başarılı bir şekilde makine öğrenme ve doğal dil işleme algoritmaları kullanılarak sınıflandırılmıştır. Bu tezde uzak homolog protein tespitin çalışmalarında, sadece pozitif örnekler değil de hem pozitif hem negatif protein sırası örnekleri kullanıldığında daha başarılı sonuçlar alındığı için çalışmamızda her iki tip protein örnekleri de kullanılmıştır.

Bu tezde, temelde SCOP 1.53 protein veritabanı çalışmada uzak homolog protein tespiti yapmak için kullanıldı. Uzak homolojileri en iyi performansla tespit etmek için önemli adımlardan biri proteinlerin özelliklerini en faydalı şekilde elde etmek, diğeri ise bu özellikleri en iyi şekilde sınıflandırmaktır. n-gram Dil Modeli kullanılarak n-gram protein dizileri çıkarıldıktan sonra, çalışmada TF-IDF ağırlıklandırma modeli ile TF-IDF Matrisi elde edilmiştir. Elde edilen protein özellikleri ile proteinleri uzak homolog ve uzak olmayan homolog olarak %95 - 99 ortalama doğrulukla sınıflandırmak için Multinomial Naif Bayes, Complement Naif Bayes, Gaussian Naif Bayes ve Bernoulli Naif Bayes olmak üzere dört tip Naif Bayes sınıflandırıcı kullanılmıştır. Bu dört tip Naive Bayes algoritması içindeki Gaussian Saf Bayes algoritması, proteinlerin uzak homolojiye göre sınıflandırılmasında en başarılı performansı %99 doğruluk ve en az zaman karmaşıklığı, ortalama mutlak hata ve ortalama kare hata değerleri ile sağlamaktadır.

Bu tezin bir bölümünde, proteinlerin yapısal sınıflandırması SCOP 1.53, SCOP benchmark ve proteinlerin kısaltılmış yapısal sınıflandırması olan SCOP 2.07'den yeni oluşturulan SCOP benchmark protein veritabanı, uzak homolog proteinleri tespit etmek için kullanılmıştır. N-gram protein dizileri, n-gram dil modeli kullanılarak çıkarıldı ve bir TF-IDF ağırlık modeli ile bir TF-IDF matrisi elde edilmiştir. İlk önce bu şekilde elde edilen özellikler, yeni inşa edilen derin öğrenme mimarisiyle sınıflandırılmıştır. Bu sınıflandırmanın sonucunda doğruluk sonuçları oldukça iyi görünmektedir. Fakat karmaşıklık matrisine bakıldığında ise az örnekli sınıf olan uzak homolog sınıfının çoğunluğunun kabul edilemez bir şekilde yanlış sınıflandırıldığı gözlenmiştir. Bunun üzerine dengesiz veri problemini ve dolayısıyla yanlış sınıflandırmayı önlemek için bir yumuşatma işlemi gerçekleştirilmiştir.

Yumuşatma işleminin ardından, elde edilen dengeli özellikler, üç SCOP veri seti ile yeni inşa edilen derin öğrenme mimarisi kullanılarak uzak homologlara başarılı bir şekilde sınıflandırılmıştır.

Bu tezde protein örneklerini yumuşatma işlemi ile edilen protein veri seti, yeni inşa edilen derin öğrenme yöntemi ile ortalama %89,13 doğruluk ve ortalama %88,39 ROC puanı ile uzak homolog proteinleri tespit edilmiştir. Bu metodun ardından uzak homolog protein tespiti için, farklı *k* en yakın komşu algoritması gibi farklı algoritmalar da denenmiştir.

Daha önce yumuşatma işleminden elde edilen bu öznitelik vektörleri farklı *k* en yakın komşu sınıflandırıcı algoritması kullanılarak sınıflandırılmıştır. Bu sınıflandırmada, uzak homolog için farklı k en yakın komşu sınıflandırıcı üzerinde Bray Curtis, Euclidean, Minkowski, Dice, Jaccard, Chebyshev, Cosine, SokalSneath, Corrrelation, Matching, RogersTanimoto, SokalMichener, Canbera, Hamming, Kulczynski ve RussellRao kullanılan farklı uzaklıklar kullanılmıştır. Ancak sadece doğruluk değerlerine bakıldığında Bray Curtis, Euclidean, Minkowski, Dice, Jaccard, Chebyshev, Cosine, SokalSneath ile KNN yöntemi ile yapılan sınıflandırmada ortalama %98,7 doğrulukla oldukça iyi bir başarı elde etmiştir. Öte yandan karışıklık matrislerine bakıldığında bu başarının tam anlamıyla doğru olmadığı görülmektedir. Bunun nedeni dengesiz veri sorunudur. Uzak homolog protein tespitinde, dengesiz veri sorununu önlemek için özel bir k-kat değeri formülü ile katmanlı çapraz doğrulama) metotu önerilmiştir. Bray Curtis uzaklığı ve özel k-kat değeri ile çapraz doğrulama ile KNN algoritmasının %99 doğrulukla en başarılı performansı gösterdiği gözlemlenmiştir.

Bu tezin uzaktan homolojiyi tespit etmedeki başarılı sonucu, yeni proteinlerin keşfinde ve dolayısıyla tıpta ilaç keşfinde çok umut vericidir. Bu çalışmada aşağıda sıralanan önemli katkıları literatüre sunmaktadır:

1) Biyoinformatikte doğal dil işleme tekniklerinin başarılı bir şekilde kullanıldığını göstermektedir.

2) n-gram işleminde doğru n-değerini seçmenin başarıyı ne derece etkilediği gözler önüne serilmektedir.

3) Dengesiz veri probleminin sınıflandırma başarısını nasıl etkilediği gösterilmiştir. Dengesiz veri problemini yenebilmek için yapılabilcek çeşitli yöntemler açıklanmıştır.

4) Bu çalışmada, sınıflandırma çalışmalarında doğruluk sonuçlarının başarısının, özellikle dengesiz veri problemlerinde tek başına yeterli olmadığı, karışıklık matrisi gibi diğer performans değerlendirme ölçütlerinin de dikkate alınması gerektiği ortaya konmuştur.

5) Yeni önerilen Derin öğrenme mimarisi, proteinlerin uzak homolojisini saptamak için yumuşatılmış veriler üzerinde iyi sonuçlar verdiği gözlenmiştir.

6) Bu çalışmada, katmanlı çapraz doğrulama üzerindeki k kat değerinin otomatik olarak farklı sınıf ve örnek değerlerine bağlı olarak belirlenmesinin önerilmesi, dengesiz veri problemlerinde oldukça faydalı olacaktır.

7)Katmanlı k-kat çapraz doğrulama ile k en yakın komşu sınıflandırıcı metodunun uzaktan homolog protein tespiti için başarılı bir yöntem olduğu gözlenmiştir.

8)Çeşitli Naif Bayes metotlarının uzak homolog protein tespitinde olan başarısı gösterilmiştir. Gaussion Naif Bayes algoritmasının başarısının da uzak homolog protein tespitinde, diğer Naif Bayes metotlarına oranla üst sırada olduğu gözlenmiştir.

9) Bray Curtis mesafesi ile KNN ve yeni k kat ile katmanlı çapraz doğrulama, uzak homolog protein keşfinde gelecek vaadeden bir yöntem olduğu gözlenmiştir.

10)Büyük veri problemlerinin moleküler biyoloji alanını da etkilen önemli bir faktör olduğu gözler önüne serilmiştir. Bu sebeple, MongoDB gibi bir NoSQL veritabanında protein sırası örnekleri depolanması, bu çalışmada gerçekleştirilerek sonraki çalışmalara ışık tutması planlanmıştır.

ACKNOWLEDGEMENTS

I am eternally grateful to my esteemed supervisor for their guidance and insights, Prof. Dr. Ulus ÇEVİK and Prof. Dr. Turgay İBRİKCİ in thesis.

I thank Prof. Dr. Turgay İBRİKCİ so much for being available whenever I need it. Furthermore, I would like to thank him for his patience and support during the times when the thesis reached a dead end.

It is a pleasure to thank to my thesis committee members and advisors Prof. Dr. Turgay İBRİKCİ, Prof. Dr. Ulus ÇEVİK, Prof. Dr. Zeynel CEBECİ, Assoc. Prof. Dr. Sami ARICA, Assoc. Prof. Dr. Lütfü SARIBULUT, and Asst. Prof. Dr. Erkut TEKELİ for valuable insight they shared and guiding advices.

I would like to thank my institution Kahramanmaras Sutcu Imam University for its support.

I would like to thank my family for their trust and support in me.

CONTENTS PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND NOTATIONS

BLAST	: Basic Local Alignment Search Tool
BMs	: Boltzmann Machines
BoW	: Bag-of-Words
FN	: False Negative
FP	: False Positive
GPCR	: G-Protein-Coupled Receptor
IDF	: Inverse Document Frequency
KNN	: K-Nearest Neighbors
LDA	: Latent Dirichlet Allocation
LSA	: Latent Semantic Analysis
MWU	: Multi-word Unit
NLP	: Natural Language Processing
NNR	: Nearest-Neighbor Rule
NoSQL	: Not Only SQL
PLSA	: Probabilistic Latent Semantic Analysis
RUS	: Random Under-sampling
SCOP	: Structural Classification of Proteins
SCOPe	: Structural Classification of Proteins — extended
SMOTE	: Synthetic Minority Oversampling Technique
SMOTE-NC	: Synthetic Minority Oversampling Technique-Nominal Continuous
SVD	: Singular Value Decomposition
SVM	: Support Vector Machine
TF	: Term Frequency
T-Link	: Tomek links
TN	: True Negative
TP	: True Positive

1. INTRODUCTION

1.1. Subject of the Thesis

In computational biology, remote homologous protein detection is among the essential problems (Chen et al., 2016; Ben-Hur and Brutlag, 2003). Recognizing the homology and remote homology of proteins provides important developments in bioinformatics because it is benefitable to discover functions and structures of proteins (Dong et al., 2006).

Algorithms based on similarity of protein sequences alone are insufficient to discover remote homologous proteins. In addition to the similarity, protein family and superfamily information have also benefited by taking into account the protein structural similarity. Proteins from the same ancestor are thought to contain similar properties. Proteins from the same family are structurally more similar than proteins from different families. Hence**,** remote homologous protein detection problems are built on the protein family and superfamily (Shah et al., 2008).

Although remote homology and similarity are different concepts, remote homology detection is established on sequence similarity. Prediction problems of remote homologous proteins can be considered as detecting protein similarity. The important point of the problems is the protein similarity ratios. If a pair of proteins are remote homologs to each other, their similarity is between 25% and 40%. On the other hand, the problem can be considered. On the other hand, the problem can be considered as detecting protein families and superfamilies. If a pair of proteins are remote homologs of each other, they are from the same superfamily but a different family.

The thesis has benefited from various fields like bioinformatics, NoSQL databases, and machine learning. Hence, basic knowledge about these fields increases the thesis's intelligibility.

1.2. Bioinformatics

Biological data on the SWISS-PROT database (Bairoch and Apweiler, 2000) and GenBank repository (Benson et al., 2009) doubles every 15 months. So, new biological data rapidly emerges. Discovering meaningful information is necessary from this immense data. Since the new computers can easily process and analyze data, processing this increasingly emerging biology data with computer algorithms is of immense attention (Luscombe et al., 2001).

Bioinformatics basically merges two fields such Molecular Biology and Genetics and Computer Science. Bioinformatics benefits from other necessary fields, such as mathematical statistics. Bioinformatics is a science branch that solves problems in molecular biology by computer science and mathematics.

Bioinformatics is composed of complex biology and complex computer fields. They think that it is a deeply hard and alarming field, before the scientists' study of the field. In fact, the fundamentals of the field are easily understood. The field analyzes symbol strings which represent protein, RNA, and DNA sequences. Bioinformatics scientists investigate string structures and string relationships to analyze the strings (Holloway, 2020).

1.3. Protein

The Scientists believe that proteins form chains of links that naturally irregularly connect a number of regions with functional functions. With mutation, changes in these regular and irregular protein regions may occur. While mutations

in irregular regions have no effect or harm; mutations in regular regions can disturb the protein's normal function (Shah et al., 2008).

Amino acids are the vital structural entities of proteins. Proteins occur by amino acids' side-by-side alignments. Amino acids are collided by side-by-side alignments of hydrogen, carbon, oxygen, and nitrogen molecules. The twenty major amino acids are in protein production. These standard twenty amino acids result in different numbers of sequences, resulting in proteins in hundreds of thousands of diverse structures. Just as you can write different words and phrases with different letters in the twenty-nine letters in the same alphabet. An infinite number different proteins can be generated using twenty amino acids. Table 1.1 shows protein sequence samples that consist of the amino acids in different orders and numbers. There are the twenty amino acids with the abbreviations that may be considered as chemical alphabets used to symbolize amino acids in Table 1.2.

Table 1.1 Samples of protein sequence

	Sample Protein Sequences
P1	QDLDEARAMEAKRKAEEHISSSHGDVDYAQASAELAKAIAQLRVIELTKK
P2	QDLDEARAMEAKRKAEEHISSSHGDVDYAQASAELAKAIAQLRVIELTKKAM
P3	QDLDEARAMEAKRKAEEHISSSHGDVDYAQASAELAKAIAQLRVIELTKKAM
P4	SFELPALPYAKDALAPHISAETIEYHYGKHHQTYVTNLNNLIKGTAFEGKSLEEIIRSSEGGVFNNAAQVWNHTFYWNCLAP
P5	VHKLEPKDHLKPQNLEGISNEQIEPHFEAHYKGYVAKYNEIQEKLADQNFADRSKANQNYSEYRELKVEETFNYMGVVLHELYFGMLTP

Table 1.2 The Standard 20 amino acids

	Single-letter Code	**Three-letter Code**	**Amino Acid Title**
1	A	Ala	Alanine
2	C	Cys	Cysteine
3	D	Asp	Aspartic Acid
4	E	Glu	Glutamic Acid
5	F	Phe	Phenylalanine
6	G	Gly	Glycine
7	H	His	Histidine
8	I	Ile	Isoleucine
9	K	Lys	Lysine
10	L	Leu	Leucine
11	M	Met	Methionine
12	N	Asn	Asparagine
13	P	Pro	Proline
14	Q	Gln	Glutamine
15	R	Arg	Arginine
16	S	Ser	Serine
17	T	Thr	Threonine
18	V	Val	Valine
19	W	Trp	Tryptophan
20	Y	Tyr	Tyrosine

1.4. Homolog Protein & Remote Homolog Protein

The Science has searched for new solutions to predict new proteins' functions in the last forty years. In well-defined databases, similar ancestral protein sequences are believed to contain related functions. The presumption is that proteins containing similar protein sequences or compositions have similar protein structures. This presumption makes it easier to extract new meaningful information and discover new proteins in increasing biological data (Shah et al., 2008).

Homolog proteins share similar structures and functions that have high sequence similarities. A homology may describe as a relation among any pair of proteins in the same family class. Proteins having a pairwise sequence identity of more than 40% similarity are homolog (Chen et al., 2018). Proteins with sequence similarity, but proteins with low similarity are remote homologous to one another (Liu et al., 2014). Remote homology is defined as homolog proteins pairs comtaining a pairwise sequence identity between 25% and 35-40%. Classification of any protein pair in the same superfamily but the diverse family has considered as a remote homology.

1.5. NoSQL Database in Bioinformatics

Depending on whether genomic sequencing science is progressing rapidly, the amount of sequence data is increasing. Hence, direct processes with amino-acid sequences are one of the considerable challenges in computational biology. For this reason, the idea of algorithm parallelizing has arisen in the computational biology field (You et al., 2014). The basic logic of parallel processing is that are divided big data into small parts and processed each part in parallel using a different computer resource (Dean and Ghemawat, 2008).

1.6. The Purpose of the Thesis

There are a couple of aims of the thesis; the main one is to develop an application that can accurately detect remote homologous protein because it is an important problem in the field of bioinformatics. The other aim, which is related to the previous aim, is to obtain fixed-length feature vectors from protein sequences and to obtain features that can be used in machine learning algorithms.

The thesis shows the similarity of protein sequences to natural languages and demontrates that natural language processing applications can be used in the field of bioinformatics. The thesis also shows how much the motif length affects the working performance of the motifs obtained for protein properties. The correct number of n defines the right motif.

Another aim of the thesis is to develop an algorithm that produces solutions to unbalanced data problems for remote homologous proteins. Having unbalanced data is usually normal in classification problems. However, this imbalance is an important problem in the finding of protein structures. Because it is quite sharp when the presence of the majority class is much higher than that of the minority class, it causes the success to be inconsistent. The application of the KNN method together with the newly proposed k-split formula is designed to accurately detect remote homologous proteins.

The Bray Curtis, Euclidean, Minkowski, Dice, Jaccard, Chebyshev, Cosine, SokalSneath, Corrrelation, Matching, RogersTanimoto, SokalMichener, Canbera, Hamming, Kulczynski, and RussellRao distance measurements are applied to have much more accurate results for the remote homologous proteins' problem because choosing the distance with an accurate method will give accurate results on such thin sequence similarities.

Finally, in the thesis, a new application that trains with a new deep learning network that balances TF-IDF feature vectors calculated over n-grams on the unbalanced dataset with smoothing operations.

1.7. Challenges of the Thesis

The difficulty of detecting remote homologous proteins originates in detecting proteins consisting of very low sequence similarity. Hence, remote homology detection is much more difficult than homology detection. The remote homology problem is a multi-featured, and multi-class problem that is difficult to solve (Damoulas and Girolami, 2008). There are four basic challenges in remote homology detection: storing, and accessing protein dataset, protein representation, cost, and efficiency.

1.8. Storing and Accessing Protein Dataset

The quantity of protein sequences is increasing quickly as sequencing technology develops. More than 64 million protein sequences have been contributed to the UniProtKB/TrEMBL database (Leinonen et al., 2006) as of June 2016, and millions more are added each month. Due to their rapid growth, these proteins need to be stored in a particular database. The relevant database is updated as new proteins are found. Therefore, a more adaptable storage method is required for the thesis data. Non-relational databases have taken the lead over relational databases in order to handle the complexity of large datasets.

1.9. Protein Representation

The two most common methods built up to discover protein similarity are the Smith-Waterman algorithm, and BLAST (Ben-Hur and Brutlag, 2003). Despite performing many applications to figure out the protein similarity problem, its solution is a difficult problem. Therefore, remote homologous protein detection is thought to be the detection of subtle sequence similarites of protein (Chen et al., 2005). Decreasing sequence similarities complicates predicting homology. Therefore, prediction of remote homology remains a major problem in computational biology.

In addition to subtle sequence similarities, there are challenges for remote homology detection such as fixed length feature vectors, unbalanced data, and misalignment for protein representation.

1.10. Cost and Efficient

While protein sequence numbers are increasing speedily, the need for calculations with low calculation costs is increasing. Diminishing the methods' cost is needed since methods to detect remote homology are time-consuming.

When classification performs with both reference datasets and an independent dataset, the new method performance will increase. In bioinformatics, new proteins are discovered over time. The performance of the predetermined methods in these newly discovered proteins is not exactly known. Furthermore, protein features is one of the important steps of remote homology detection. Automatically discovering these features provides the thesis performance improvement. Natural Language Processing (NLP) algorithms provide the process in this thesis.

2. RELATED WORKS

Discovering similar aminoacit sequeunces is an interesting field in bioinformatics. Protein family and sequence similarities have both surprised and piqued the interest of researchers. These discoveries appear to support the opinion that ensures information about the evolutionary, structural, and functional properties of proteins (Radivojac, 2022).

There are four categories: aligment methods, genarative methods, discriminative methods, and ranking methods in applications to detect remote homology. Some trade-off methods based on alignment methods such as the Smith–Waterman algorithm, Basic Local Alignment Search Tool (BLAST), Position-Specific Iterated BLAST (PSI-BLAST), and Fast Approximation to Smith–Waterman (FASTA) have been proposed to improve efficiency (Altschul et al., 1997; Altschul et al., 1990; Pearson, 1991) in protein processing methods. Homologous proteins are quickly identified by the alignment of functional local regions in proteins (Shah et al., 2008). In the initial applications, pairwise alignment methods such as local and global alignment have been applied to find protein remote homology (Needleman and Wunsch, 1970; Zou et al., 2015). But when there is low sequence similarity, therefore, in the exploration of remote homologous proteins; aligment-based methods cannot show the desired success when applied directly (Rost, 1999).

Generative algorithms after pairwise alignment methods have been performed for the detection of remote homology. The generative algorithms have increased the accuracy estimation for remote homology. Positive samples of a protein family or superfamily in generative models have been trained. Karplus et al. (1998), form a profile hidden Markov model. They searched in a large database

(Karplus et al., 1998). Nowadays, discriminative methods are improved rather than derived from pairwise algorithm and generative methods. The discriminative methods take information from positive and negative protein samples and retain the protein sequence characteristics (Chen et al., 2018). Using fixed length feature vectors is required to perform discriminative method (Chenet al., 2016).

Using profile-based protein sequences rather than only by amino acid sequence information provides better performance according to recent studies (Liu et al., 2013; Lingner and Meinicke, 2008; Liu et al., 2013; Liu et al., 2014). Based on the average word similarity, the word correlation method identifies the characteristic regions in protein sequences with high efficiency (Lingner and Meinicke, 2008).

The amino acid sequences are applied to determine remote protein homology. The amino acids are considered words. Each amino acid is symbolized by a letter. Hence the natural language processing methods are benefical to calculate remote protein homology in the respect (Chen et al., 2018). Computational approaches to detect protein remote homology save cost compared to the traditional biological techniques. They are more efficient (Chen et al., 2018).

Liu et al. (2014) predicted a family of proteins from other families on the SCOP Database using protein motifs as an SVM kernel. According to the paper, an SVM kernel with protein motifs is more successful than a kernel with BLAST or Smith-Waterman. It is to classify by the SVM algorithm combining protein motifs and kernel measure, which is more successful than classifying by the k-nearest neighbor algorithm (Liu et al., 2014). A kernel can be defined to show a sequence similarity as a dot object in a space. Methods based on kernels are suitable for sequential objects (Kiros et al., 2015; Li et al., 2015; Farabet et al., 2013). There is

evidence that the mismatch kernel has better performance than the Fisher kernel in remote homology detection (Kiros et al., 2015).

Since kernel-based methods evaluate N^2 kernel functions for N sequences, large-scale data computation becomes problematic. The computation is quite expensive as kernel computations are required between test and train samples. Hence, they have poor interpretation and expensive estimation, although discriminative kernel-based models are successful with high accuracy (Lingner and Meinicke, 2008).

Since remote homolog protein detection is a multi-class problem, it has engaged attention in pattern recognition solutions. Many protein features, such as protein secondary structure and amino-acid composition, are detected in remote homology detection. In addition, there are the features derived by sequence alignment methods. So, a multi-featured dataset is born for remote homology detection. The fact that the data is multi-featured also sets an example for pattern recognition and other multi-featured problem solutions. Since the protein dataset used in remote homology detection is a multi-featured dataset, Damoulas and Girolami developed a new multi-class kernel that offers the opportunity to evaluate different feature groups together in 2008. The proposed method successfully performed remote homologous protein detection with a ROC value of 92.4% by using both global protein characteristic features and features obtained by sequence alignment algorithms using a variational Bayes approximation (Damoulas and Girolami, 2008).

Different academic disciplines, such as natural language processing and text categorization, have benefited from classifying protein. In the thesis (Chorowski et al., 2015), patterns as words are extracted from protein sequence language using the TEIRESIAS algorithm. Efficient features are selected from the patterns using the

Chi-square algorithm. A vector composed of selected patterns' occurrence times is created from each protein sequence. This new representation of the protein sequences is classified by the SVM algorithm. This method, which is called the SVM-pattern, has been more successful than BLAST (Altschul et al., 1990), and SVM-based methods such as SVM-k-spectrum (Kiros et al., 2015), and SVM-pairwise (Leslie et al., 2001) to detect remote homology (Chorowski et al., 2015).

SVM-based methods to detect remote homology suffer from the peaking phenomenon due to large and noisy data. Hence, the thesis (Liu et al., 2013) focuses on extracting protein features and representing proteins efficiently rather than explicitly representing proteins. In this thesis, unlike representing protein sequence by n-grams, patterns, or motifs, protein sequences are created as documents with bags-of-words by the latent semantic analysis (LSA) method. Then these protein sequences representations are classified using SVM. The study is valuable since the word-document matrix to represent protein sequence is formed for the first time. The study (Liu et al., 2013) has better performance than kernel and sequence-based methods such as SVM-LA (Eskin et al., 2003), PSI-BLAST (Altschul et al., 1997), and SVM-pairwise (Leslie et al., 2001).

A significant first step toward improving protein remote homology detection is to incorporate evolutionary information into the profiles of proteins. In the study (Liu et al., 2013), a new protein representation is based on extracting the beneficial evolutionary information in frequency profiles.

In the study (Lovato et al., 2015), protein structure was benefited in addition to sequence-based approaches to detect remote homology. The study is a multimodal approach since it uses both protein sequences and protein 3D structures for remote homology detection. GPCR superfamily analysis shows that there is no information

that can be extracted by only sequence-based approaches in the study (Lovato et al., 2015).

ProtDec-LTR (Liu et al., 2015) is set on protein sequences to detect remote homology. ProtDec-LTR has more sensitive and more stable performance advantages than studies such as disPseAACi, SVM-Ensemble, PDC-Ensemble, and dRHPPseRA to detect protein remote homology since it is the first approach to merge ranking predictors using Learning to Rank (LTR) (Liu et al., 2015). ProtDec-LTR2.0 is formed by adding the evolutionary information from representations of pseudo proteins into ProtDec-LTR. The dataset from SCOPe with more protein samples is used to train ProtDec-LTR2.0 for the purpose of being more useful. ProtDec-LTR2.0 facilitates detecting proteins' sequence and structure information since it has a web server that has a graphical interface (Chen et al., 2017).

Shao et al. (2021) have performed a fusion study combining both classification and network-based methods to decrease the false positive rate in remote homology detection. The method is named ProtRe-CN, which is a ranking method via combining both LinearSVM methods and four network methods. ProtRe-CN method has extremly succeed with 93.27% AUROC score because the method has strengths of both classification and network methods (Shao et al., 2021).

3. MATERIALS AND METHODS

3.1. SCOP Database

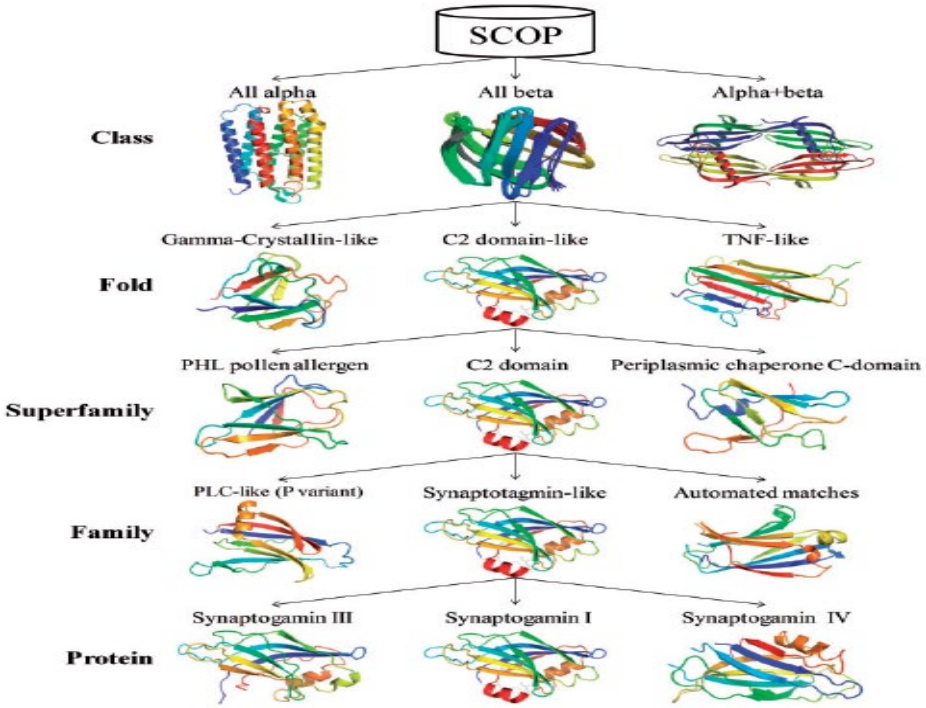

Figure 3.1. SCOP database hierarchical structure (Chen et al., 2018)

SCOP, which is built on the tree hierarchy, is a database of protein structure classification as presented in Figure 3.1. In SCOP, proteins are categorized into four levels: families, superfamilies, folds, and classes (Murzin et al., 1995). The class level, which keeps information about protein secondary structure, is the highest level. Fold which keeps fold structure of protein is second level. The fold level keeps superfamilies similar. The superfamily level, which keeps the proteins whose

sequences are similar by less than 30%, is the third level. Family which is forth level in the SCOP database.

The SCOP database is based on their functional and structural similarities. Generally, proteins of the same family have an obvious evolutionary relationship. Remote homologous proteins are within different families. They arise from the same superfamily. Proteins with the same fold but without the same superfamily have an unobtrusively evolutionary relationship. The proteins within different classes does not have evolutionary relationships (Murzin et al., 1995; Andreeva et al., 2004; Chen et al., 2018; and Chandonia et al., 2018).

3.2. Protein Datasets in the Thesis

In this thesis, proteins obtained from the SCOP database (SCOP), which is a database that includes protein relationships, which are extensively performed in many studies, have been used in the detection of family, superfamily, and remote homologous proteins. The SCOP 1.53 version protein has been used primarily and mainly, and trials have made use of detecting family, superfamily and remote homologous proteins in the thesis. The SCOP 1.53 benchmark data-set contains 4352 proteins from 54 families.

The first dataset in the thesis; SCOP database, is known as a gold protein database. SCOP 1.53 protein database which is abbreviated Structural Classification of Proteins is made use of detecting remote homologous proteins in the thesis. 4352 protein sequences are taken from the SCOP 1.53 protein database to explore remote homolog proteins. The protein dataset belongs to 1356 different superfamilies and 853 different families.

The detection of remote homologous proteins in the current thesis has been performed with the SCOP database. Herein, 3 different datasets from the SCOP

database have been used, 2 of which have been used previously to detect remote homolog proteins; SCOP 1.53, SCOP benchmark, and the newly created SCOP benchmark dataset from SCOPe 2.07.

Families containing at least 10 homolog proteins and at least 5 superfamilies outside of the superfamily of the target family have been chosen from the SCOP 1.53 protein database as 54 families. Next, 4352 protein sequences have been received from the SCOP 1.53 protein database to explore remote homolog proteins. The e-value of the pairwise alignments of these protein sequences taken from the Astral database is not higher than 10^{-25}. These proteins belonged to 1356 different families and 853 different superfamilies. The SCOP benchmark dataset, comprising 102 target families, has been taken from the SCOP database and has similar to the SCOP 1.53 dataset. The SCOP 2.07_v1 benchmark dataset comprises 51 target families, which have been taken from the SCOP database from SCOPe 2.07.

Families containing at least 5 homolog proteins and 5 superfamilies outside of the superfamily of the target family have chosen from the new SCOPe 2.07_v1 protein data version as protein sequences with less than 40% identity to each other. The fundamental building blocks of the protein sequences are amino acids. Although there are more than 700 types of amino acids in nature, only 20 major amino acids are used in protein production, as shown in Q = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y} (Wu et al., 2014).

3.3. NoSQL Database

NoSQL databases resolve big data when relational databases do not overcome the data. NoSQL is an abbreviation of "Not Only SQL". NoSQL unstructuredly copes with the data, unlike conventional relational databases. In general, NoSQL databases have a non-relational based structure. NoSQL databases

have a several technologies such as MongoDB, Neo4j, and Cassandra etc. (Shao and Conrad, 2015).

In the field of bioinformatics, it is an important research topic to store large amounts of data themselves. In addition, ensuring the performance of these newly obtained data is an important research topic while performing the workflow (Guimaraes et al., 2015). In the study, MongoDB, which is a document-oriented NoSQL database, is used in order to store data in bioinformatics. In this study, a real MongoDB application is implemented and its advantages and disadvantages are shown. MongoDB among document-oriented NoSQL databases such as ArangoDB, Elastic Search, and OrientDB databases is preferred in the study, since MongoDB has achieved more successful performance with a higher rate (Mishra et al., 2018).

3.3.1. MongoDB Database

MongoDB has improved using C++ with 10gen in 2009. It is a non-relational database with open source code It has high performance (Shao and Conrad, 2015). MongoDB is a based database on document. MongoDB is an independent schema. MongoDB is simply scalable because it has immense queries and fast updates. It undifficulty performs data insertion, deletion, and update processes. The documents in MongoDB are stored as BSON format. MongoDB provides features such as aggregation, indexing, replication, consistency, fault tolerance, auto shading, persistence, and high availability (Shao and Conrad, 2015; Öztürk and Atmaca, 2017; http://www.buyukveri.co/category/mongo/; Aggarwal and Roopam, 2016). During data update operations, the database can read the reader data as long as the database is not locked.

3.3.2. MongoDB Usage in the Thesis

The PyMongo 3.7.2 version which is the official MongoDB driver in Python, is used to improve non-relational databases in the study (api.mongodb.com). The MongoDB database's preference reasons in this thesis are that it is document-based and is independent of the schema. Dataset samples are text at different lengths. Documents are stored in JSON format in MongoDB. The JSON format is not a table format in relational databases. JavaScript Object Notation, which is abbreivated JSON is improved from the JavaScript language (Singh Chauhan et al., 2015). It is an independent and interchangeable text format to store document files.

Table 3.1. Protein documents on MongoDB database

{
id:"1",
name:"d3sdha_ ",
superfamily:,” 1.1.1”
family:”1.1.1.1”
sequence:”
svydaaaqltadvkkdlrdswkvigsdkkgngvalmttlfadnqetigyfkrlgnvsqgmandklrghsitlmya
lqnfidqldnpddlvcvvekfavnhitrkisaaefgkingpikkvlasknfgdkyanawaklvavvqaal“
status:"1"
}
{
id:"2",
name:”d1b0b__”
superfamily:" 1.1.1",
family:” 1.1.1.1”
sequence:”
lsaaqkdnvksswakasaawgtagpeffmalfdahddvfakfsglfsgaakgtvkntpemaaqaqsfkglvs
nwvdnldnagalegqcktfaanhkargisagqleaafkvlagfmksyggdegawtavagalmgmirpdm”
status:"1"
}.

...

{
id:"N-1",
name:" d1hdj",
superfamily:” 1.2.2”
family:” 1.2.2.1”
sequence:”
mgkdyyqtlglargasdeeikrayrrqalryhpdknkepgaeekfkeiaeaydvlsdprkreifdrygeeglkgs
gc” status:"1"
}
{
id:"N",
name:"d1qbha_",
superfamily:” 7.52.1”
family:” 7.52.1.1”
sequence:”
gshmqthaarmrtfmywpssvpvqpeqlasagfyyvgrnddvkcfccdgglrcwesgddp
wvehakwfprceflirmkgqefvdeiqgryphlleqllsts”
status:"1"}

3.4. Preprocessing and Feature Extraction Algorithms

FASTA sequences of proteins are read using the Biopython tool (https://biopython.org/) in the Python programming language.

3.4.1. Bag of Words Model

Texts are converted into a numeric feature vector to be processed in NLP. A widespread model for converting texts into numeric vectors is the Bag of Words (BoW). A fixed-length feature vector is needed for NLP. BoW also generates a solution for the fixed-length feature vector. (Zhao and Mao, 2017).

In the bag of words model, documents in a dataset are turned into feature vectors based on all the words in all documents. A bag is basically a dictionary vector composed of all unique words in all documents in a dataset. By analyzing existing documents, a bag containing each unique word is obtained. Depending on the bag, a feature vector is created for each document sample. The BoW model does not take cognizance of the word order in documents. It calculates the number of times the word “bag” occurs in the document. If the document does not include the word, zero represents the word on the feature vector. If the document contains the word, the occurrence number in the document is representated as the word on the feature vector. Hence, for each document, a feature vector in the created bag length is created, and thus with a fixed length is created (El-Din and Doaa Mohey, 2016).

In addition to NLP, BoW is one of very often performed data representation techniques in many severe fields, such as object categorization (Zhang et al., 2010).

3.4.2. n-gram

An n-gram, which is also called a multi-word unit (MWU), is an n-length sequence of slices of an item such as a number, digit, word, or letter etc. According to n-gram, the sequences with 1 length of an item are called unigrams or monograms, the sequences with 2 lengths of an item are called as bigrams, the sequences with 3 length of an item are called as trigrams.

In the study, n-gram is explained over text since the data is text that is composed of amino acids. Therefore, the data is protein sequences and n-grams are amino acids or amino acid sequences according to n number. In the n-gram algorithm, parts with a width of n are obtained from the served texts. This method determines the number of times in these n-length pieces' each text. Depending on the n value, the probability space that the array will take varies (Bayrak et al., 2012).

3.4.3. TF-IDF Weighting

TF-IDF Weighting has been performed to find out relations among terms with the document. The objective is exploring the term's conceptual meaning in this document and acquire information regarding the sample document. TF-IDF Weighting is frequently utilized for text mining. Term frequency (TF) is the measure of a term occurrences number in a document. If a term is used more often than others in the document, it means the term is more interrelated to this document. When calculating TF; all terms are hypothesized to have equal importance. Certain terms such as "is", "the", and "of" are of minor importance. Hence, IDF is a measure of term importance in the sample document (Qaiser and Ali, 2018).

TF-IDF Weighting is obtaining a TF-IDF Matrix, which is created by multiplying TF and IDF values. Equation 1 shows the common TF-IDF weighting

formula, which consists of TF being abbreviated term frequency and IDF being abbreviated inverse document frequency (Zhang et al., 2011).

$$TF - IDF\ Weight = TF \times IDF \quad (1)$$

$$TF = tf_{i,j} \quad (2)$$

$$IDF = log\left(\frac{N}{df_i}\right) \quad (3)$$

$$w_{i,j} = tf_{i,j} \times\ log\left(\frac{N}{df_i}\right) \quad (4)$$

In the Equations 1,2,3 and 4, $w_{i,j}$ is the weight of n-gram i in protein sequence j, N is the protein sequences number in the dataset, $tf_{i,j}$ is the n-gram frequency of trigram i in protein sequence j and df_i is the protein sequence frequency of n-gram i in the dataset, in consequence of n-grams are terms, protein sequences are documents and the protein dataset is collection in the study.

3.4.4. Latent Dirichlet Allocation (LDA)

Blei et al. developed the LDA which is a probabilistic model for discrete data. LDA is a Bayesian model with a three-level hierarchical, LDA models each member as a topic forming the basis of a finite mixture. LDA has been applied many non-textual areas, such as bioinformatics, collaborative filtering, and content-based

image retrieval. Although the LDA is not only used depending on the text, it is studied based on the text in the study. Therefore, LDA in the study is explained based on text (Blei et al., 2003; Endres et al., 2009).

In LDA based on text, each sample is a document. Each document composed of words. Features of samples are each word existence and clusters are topics. The aim of LDA is to extract topics contained in documents. LDA presumes that each document composed of multiple topics. LDA ignores the relationship between the topics. LDA builds each document as a discovered topics' distribution (Blei et al., 2003; Endres et al., 2009).

3.4.5. Probabilistic Latent Semantic Analysis (PLSA)

The aim of the Latent Semantic Analysis (LSA) that is a notable feature extraction method especially in natural language processing is to extract the hidden topics or semantic relationships.

PLSA, which is abbreviated as Probabilistic Latent Semantic Analysis, is a feature extraction method, especially for natural language processing as in LSA. PLSA, which has emerged by evaluating LSA from a statistical and probabilistic perspective, is a method to extract hidden topics (Jarad et al., 2015). PLSA does not perform Singular Value Decomposition (SVD) for hidden topics extraction. PLSA extracts the dataset's hidden topic with the probability method. The topic anlaysis model supposes that the documents are formed by combining of some hidden topics, and the topics are formed by combining of some collection of words.

3.4.6. Regularization

One of the elementary problems that are worked on effectively in machine learning problems is to work on only training data that will perform well. The changes in this learning algorithm, which are made to build up an algorithm that will provide performance well both training data and on new data, are generally called regularization. These regularization methods are established despite increasing training errors.

The biological sequence is similar to NLP in recent years. As a result, NLP methods have benefited on bionformatics topics (Dong et al., 2006). n-grams derived from all genome protein sequences are investigated and then statistical attributes are derived from these n-grams (Dong et al., 2006; Ganapathiraju et al., 2002). Hence, n-grams that are taken from protein sequences are used to find remote homologous proteins in the thesis.

G-Protein-Coupled Receptor (GPCR) sequences dataset has classified by Decision Tree and Naive Bayes classifier. The dataset features extracts with chi-square and n-gram techniques in the study (Tripathy and Rath, 2017). The GPCR dataset is classified using Decision Tree algorithm by 1-grams, 1 and 2-grams and 1,2 and 3-grams with 89.4%, 89.5% and 89.3% accuracy, respectively. At the same time, GPCR dataset is classified using Naive Bayes algorithm by 2-grams, 2 and 3-grams, 2, 3 and 4-grams and 2, 3, 4 and 5-grams with 80.7%, 96.3%, 95.6% and 94.8% accuracy, respectively. The study shows that Naïve Bayes algorithm more successful than Decision Tree algorithm for classification of protein. However, the study also proved the effect of determining the correct n number in n-gram on the algorithm success. (Cheng et al., 2005). Naive Bayes classification algorithm executes classification for many areas such as text categorization, and medical diagnosis successfully (Bouckaert, 2004; Furat and Ibrikci, 2017; McCallum and

Nigam, 1998; Tripathy and Rath, 2017). The variables of Naive Bayes are independent. Covariance matrix calculation does not require in Naive Bayes. So, the Naive Bayes classification algorithm has a benefit, which is its usage to test a of the dataset for classification (Palaniappan and Awang, 2008; Pattekari and Parveen, 2012). These advantages of the Naive Bayes classifier and the achievement of previous studies led to Naive Bayes classifier testing in the thesis.

3.5. Methods for Unbalanced Data Problem

When the training samples dispersed at very different rates among the different classes, the unbalanced data problem appeared. The unbalanced data is that training data's one class's sample number is much less or much more than the other classes' samples' number (Jiang et al., 2013). It is characteristic of an unbalanced data problem in a binary classification that the positive training sample number is immensely less than the negative training sample number in the remote homology problem.

Resampling techniques are used to solve unbalanced data problems. The subcategories of these techniques, oversampling, undersampling, and hybrid methods by combining them, are clarified in the below subparts of 3.5 parts. (Zhihao et al., 2019; Longadge and Dongre, 2013).

3.5.1. Random Oversampling

The oversampling is a fairly simple and easy technique that tries to balance classes by replicating the samples of the minority class. The method does not cause information loss. The method can cause overfitting due to information duplication.

3.5.2. SMOTE

In 2002, Chawla described the Synthetic Minority Oversampling Technique (SMOTE) algorithm. SMOTE is an algorithm to produce new samples in the minority class to break the unbalanced in the dataset. It generates new synthetic samples for the lesser class instead of multiplying existing samples to not cause overfitting. The SMOTE algorithm main working principle is as follows:

I. It selects samples closest to the feature space.

II. It draws a line in this feature space among these selected closest samples.

III. It creates new samples by selecting new points on this line.

3.5.3. BorderLine SMOTE

While the oversampling method creates new samples by copying the minority samples, it causes data abundance. Therefore, it needs to perform less amount but more useful tasks. The BorderLine SMOTE method removes the need for too much data (Smiti and Soui, 2020).

The technique is designed by bordeline between the unbalanced class samples and the SMOTE technique. Samples on or near the borderline have more possibilities for misclassiffication than the far-borderline samples. The BorderLine SMOTE technique aims to advance the classification performance by only creating samples close to the borderline (Han et al., 2005).

3.5.4. SMOTE-NC

SMOTE-NC is an oversampling technique for nominal features based on SMOTE. SMOTE-NC is created to cope with the problem that the SMOTE technique cannot process datasets with all nominal features. It is a method that treats nominal features as distinct from continuous features. This method preserves the

features' original labes in the reproduced samples. However, the method needs to repair its aspects of failure of multi-label features (Ganganwar, 2012).

3.5.5. ADASYN

The ADASYN method has been born with the idea of multiplying different samples number depending on the diverse distribution of different minority class samples. The minority class learning difficulty level determines this distribution. More samples are reproduced from minority classes in the method. The minority class's discovery is harder than the majority class's discovery. While this method reduces the prejudice brought about by class imbalances; classification shifts the decision boundary towards difficult classes or even difficult examples. (He et al., 2008).

3.5.6. Random Undersampling

The undersampling techniques reduce the training set size to decrease the computational load. These methods may also cause valuable information to be lost. The random undersampling method is an undersampling method that takes out random samples from the majority class. Thus, the method equalizes the minority and the majority class samples' number.

3.5.7. Cluster

Because the undersampling methods cause information loss, Yen and Chen have propounded a cluster-based undersampling method (Yen and Lee, 2009). This method's idea is to preclude the training set from tending to the majority class by separating k different clusters. Since each cluster will contain its own unique features, different clusters are created. Whichever of the majority or minority classes

of a cluster has the greater number of samples, the results demonstrate a trend towards that sample class. In order to avoid this class orientation, the majority class samples' appropriate number is selected in each cluster.

3.5.8. Tomek Links (T-Link)

T-Link is an undersampling method designed by developing the Nearest-Neighbor Rule (NNR) (Thai-Nghe et al., 2000). If two different samples from different classes are determined as each other's nearest neighbors, this pair of samples is defined as T-Link.

Besides being used as the T-Link undersampling method, it can also be performed as a data cleaning method. When T-Link is used to perform undersampling, majority class samples are eliminated. When T-Link is used to clear data, eliminations are made from all classes.

3.6. K-foldCross Validation

Cross validation is a resampling method. It is a statistical method. It assesses machine learning algorithms more accurately. It is widely performed because it's simple to perform and understand. It has lower overfitting and bias. The cross validation method's several types are the Jack Knife test, Monte Carlo test, disjoint sets test, bootstrapping, and three-way split test (Saud et al., 2020).

The cross validation method gave inconsistent results for a low number of data points. The method is repeated for all processes to prevent the problem. Inspired by the idea of repeating cross validation, the idea of k-fold cross validation has emerged (Braga-Neto and Dougherty, 2004). The cross validation method with k parameters, is named k-fold cross validation. The k value shows the number of the dataset groups. The k parameter shows the algorithm repeat number in the k-fold

cross validation method. K data groups are composed of an equal number of samples. For k data groups, the algorithm is performed separately. While samples' one group among the k subgroups are retained as test data, the remaining k-1 subgroups are processed as train data. These k algorithm tests' average results are admitted to the whole dataset's result.

This model's success originates from each sample's usage at least once in both training and validation of the model. The k value choice is important in this method. Because if the k value is chosen high, the variance reduces but the computational cost rises. The low choice of k value reduces the computational cost. However, a high variance value emerges in the results. (Hagan et al., 1996). While the most popular k values range from 2 to 10, 10-fold cross validation is frequently performed in studies. It is shown that the balance of computational complexity and reliable variance range is best achieved in 10-fold cross validation by Kohavi (Kohavi, 1995). Computational complexity is the method disadvantage. The k-fold cross validation method works very slowly with nonlinear models due to the computational complexity.

3.6.1. K-fold Stratified Cross Validation

Stratified cross validation (Stratified CV) is a method developed from CV. There must be the same average response value in all folds for Stratified CV. The dataset in k-fold Stratified CV has been distributed according to a specific rule into k folds. The special rule found in the Stratified CV is to carefully maintain the proportions of the classes' the sample distribution. However, there are ignored class distribution rates in the normal CV method (Zeng and Martinez, 2000).

3.7. Similarity Measurement Methods

The distance among vectors is a beneficial parameter to calculate the vector similarity. Various similarity measurement methods are used to calculate the distance. Some methods of similarity measurement are explained in this section. It is assumed that x and y are the vectors to calculate the similarities in this section.

3.7.1. Cosine Similarity

Cosine similarity commonly measures similarities between two vectors. The cosine similarity is commonly utilized on lots of fields as document classification, intrusion detection, and recommender systems (Al-Anzi and AbuZeina, 2017; Kumar, et al., 2015; Schwarz et al., 2017).

The cosine similarity measure is described by Theodoridis and Koutroumbas in 2008 as seen in Equation 5 (Theodoridis and Koutroumbas, 2008). Cosine similarity is also called the angle cosine or angle between two vectors.

$$S_{cosine}(x, y) = \frac{x^T y}{\|x\|\|y\|} \quad (5)$$

Cosine similarity is calculated in the range of -1 and +1. Whenever the measure result approaches -1, it means less similarities between the vectors. On the other hand, it means more similarities as it approaches +1.

3.7.2. Euclidean Distance

The Euclidean distance is a simple and widespread method of measuring the distance between two points associated with the L2 vector norm, as shown in Equation 6. It is not scale dependent. Depending on the unit properties, it can cause

skewness. As the data dimensionality increases, the usefulness of the Euclidean distance decreases.

$$E\_D = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2} \quad (6)$$

3.7.3. Bray Curtis Distance

The Bray Curtis distance is also called the Sorensen distance. The distance sees space as a grid, as shown in Equation 7. The Bray Curtis distance ranges between 0 and 1 if each coordinate is positive. If the distance is 0, both coordinates represent exactly the same coordinate (Al-Hassai and Kalyankar, 2013).

$$Bra\_d_{=} \sum_{i=1}^{n} \frac{|x_i - y_i|}{(x_i + y_i)} \quad (7)$$

3.7.4. Chebyshev Distance

Euclidean distance is an effortless and widespread method. The maximum value distance names for Chebyshev distance, because it is calculated with vector spaces containing the largest distance between two vectors. The distance formula is as shown in Equation 8 (Religia and Sunge, 2019). It is a suitable distance method for samples that differ in any one dimension.

$$Che\_D = {}^{n}_{i}max\ (x_i - y_i) \quad (8)$$

3.7.5. Dice Distance

Dice distance is a distance metric generated from dice similarity. Dice distance is calculated by subtracting the dice similarity by 1, as shown in Equation 9 (Prasath et al., 2017). The distance may be suitable for values close to 0.

$$D\_D = 1 - \frac{2\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2} \quad (9)$$

3.7.6. Hamming Distance

Richard Hamming defined Hamming distance in 1950. The method calculates two vectors' difference with equal length, as shown in Equation 10 (Stabili et al., 2017).

$$H\_D = \sum_{i=1}^{n} |x_{i-} y_i| \quad (10)$$

3.7.7. Jaccard Distance

The Jaccard distance is got by subtracting the jaccard index by 1, as shown in Equation 11. Therefore, it figures out the samples' difference. According to the Jaccard method, if the sample similarity increases, the Jaccard distance is close to 0. Contrarily, if the sample similarity decreases, the Jaccard distance is far from 1 (Cha, 2007).

$$J_D = \frac{\sum_{i=1}^{n} (x_i - y_i)^2}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} x_i y_i} \quad (11)$$

3.7.8. Kulczynski Distance

The Kulczynski distance is also called Quantitative Symmetric Dissimilarity (QSK) (Kocher and Savoy, 2017). The Kulczynski distance provides results unlike those from standardized distances. The Kulczynski distance formula is shown in Equation 12.

$$K\_D = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} \min (x_i, y_i)} \quad (12)$$

3.7.9. Matching Distance

Matching distance statistically computes similarity between binary data samples (Han et al., 2011). It is supposed that A, B, C, and D are binary values where both samples have the value 1, where first sample has the value 1, other has value 0, where the first sample has the value 0, the other has the value 1, and where both samples have the value 0, respectively. In accordance with these suppositions, the matching distance among two samples is shown in Equation 13.

$$M\_D = \frac{A+D}{(A+B)+(A+C)} \quad (13)$$

3.7.10. Minkowski Distance

Minkowski distance transforms a general distance into many distances, such as Hamming and Euclidean distance, as shown in Equation 14 (Mousa and Yusof, 2018). In the Minkoski distance formula shown in Equation 14, when the *p* value is taken as 2, the equation calculates the euclidean distance.

$$Mink\_D = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p} \quad (14)$$

3.7.11. RogersTanimoto Distance

In 1960, Rogers and Tanimoto performed the Rogers and Tanimoto distance. The A, in the distance Equation 15, is the binary value where both examples have the value 1. The B, in the distance formula, is also the binary value where the first sample has the value 1 and the other sample has the value 0. The C is a binary value where first sample has value 0, other has value 1. The D is a binary value where both samples have the value 0. According to these suppositions, the RogersTanimoto

distance among two samples is shown in Equation 15. The distance similarity measure is between 0 and 1.

$$RT\_D = \frac{A+D}{A+2*(B+C)+D} \quad (15)$$

3.7.12. RussellRao Distance

The RussellRao distance was developed by Russell and Rao in 1940 (Chay et al., 2010). The distance is based on the inner product. The distance is a binary distance since matches and non-matches are served with equal weight. The distance is supposed to be A, is binary value of where both samples have the value 1. The distance is assumed n, where n is the sample number. This distance formula is shown as Equation 16.

$$RR\_D = \frac{A}{n} \quad (16)$$

3.7.13. SokalMichener Distance

These are supposed that A,B,C, D are binary values where both samples have the value 1, where first sample has value of 1, other has value 0, where the first sample has value of 0, other has value of 1 and where both samples have the value of 0, respectively. In accordance with these suppositions, the SokalMichener distance among two samples is shown in Equation 17.

$$SM\_D = \frac{A+D}{A+B+C+D} \quad (17)$$

3.7.14. Canberra Distance

In 1966, Godfrey N. Lance introduced the Canbera distance. Then, William T. Williams developed this distance in 1967. Canbera distance usefully calculates the distance among points near the origin, as shown in Equation 18 (Faisal and Zamzami, 2020).

$$Canb\_d_{=} \sum_{i=1}^{n} \frac{|x_i - y_i|}{(x_i + y_i)} \quad (18)$$

3.7.15. SokalSneath Distance

Sokal and Sneath were built SokalSneath distance in 1963. The distance has accepted both positive and negative matches (Holmberg and Hallander, 1973). These are supposed that A,B,C, D are binary values where both samples have the value 1, where first sample has value of 1, other has the value of 0, where first sample has value of 0, other has the value of 1 and where both samples have the value of 0, respectively. In accordance with these suppositions, the SokalSneath distance among two samples is shown in Equation 19.

$$SS\_D = \frac{A+D}{(B+C)} \quad (19)$$

3.7.16. Correlation Distance

Correlation distance assumes two samples' similarity if two samples' features have high relation. Although the Euclidean distance observes values far away, the Correlation distance continues this similarity assumption. The distance measures the distance that has a value between 0 and 1. The method is a version of the Pearson distance. The correlation distance among the two samples is calculated as shown in Equation 20 (Yan et al., 2015).

$$Corr\_S = \frac{x^T y}{||x||||y||} \quad (20)$$

3.8. Naive Bayes Classifier

Naive Bayes classifier is also called the “Naive Bayes Assumption”. Classifiers based on Bayes' rule have become a popular method because of their good performance on classification by presenting a concrete probabilistic model (McCallum and Nigam, 1998). Duda and Hart invented Naive Bayes classifier and first description of the Naive Bayes classification method was in 1973 (Kim et al., 2002). The method is a supervised learning classification method that is performed to classify new samples using labeled training samples. The method computes the class probability of each sample (Furat and Ibrikci, 2017; Han et al., 2011; Bhavsar and Ganatra, 2012; Duda and Hart, 1973). The probability and statisticality of this classifier come from the Bayes Theorem. Bayes theorem calculates the P(c|x), posterior probability using P(x|c), P(c) and P(x), and as shown in Equation 21.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (21)$$

where P(c|x) is the class posterior probability, class prior probability is P(c), the likelihood is P(x|c) that is the predictor served class probability and the predictor prior probability is P(x).

In the method, it is supposed that the features belonging to the sample are conditionally independent from the class. The assumption that gives this classifier the Naive label is a predictor(x) value effect on a served class (c) is independent of other predictors values (Furat and Ibrikci, 2017, Han et al., 2011, Bhavsar and Ganatra, 2012, Duda and Hart, 2012). When the features number is high, the method's performance has increased. Because the method assumes the feature's

independence. Since features are words in document classification, and these words can be quite a large number, feature data in document classification can be quite large. The method, which easily overcomes data classification with the features' large amount, is also very successful in document classification.

When the dataset of classification has n features, an 'n' dimensional feature vector is X: (x1, x2, x3… xn).

Supposed that there are 'm' classes: C_1, C_2, C_3…C_m.

The method calculates each class ci's probability of each sample X in as shown in Equation 22 and Equation 23.

$$P(x\ /c_i) = \prod_{k=1}^{n} P(x_k/c_i) \tag{22}$$

$$P(x/c_i) = P(x_1/c_i) \times P(x_2/c_i) \times \ldots \times P(x_i/c_i) \tag{23}$$

3.8.1. Bernoulli Naive Bayes (BNB) Algorithm

Various Naive Bayes classification algorithms have been developed to classify text data quickly and easily. The first Naive Bayes algorithm developed for this purpose is the Bernoulli Naive Bayes model (Jiang et al., 2013). The Bernoulli Naive Bayes algorithm stands by the BNB algorithm which is called the Multivariate Bernoulli Naïve Bayes too. The algorithm generates a document by performing Bernoulli experiments. These Bernoulli experiments form the feature vectors to consist of 0 and 1 of s vector, that is depending on whether each word is passed at least once in the document (Mendez et al., 2008). The Bernoulli Naive Bayes calculation is shown in Equation 24.

$$P(x/c_i) = \prod_{k=1}^{n} P(w_k/c_i)^{x_k} \times \left(1 - P(w_k/c_i)\right)^{1-x_k} \tag{24}$$

where x is a vector that includes word numbers belonging each sample, c is a vector that includes class of each sample.

3.8.2. Multinomial Naive Bayes (MNB) Algorithm

The MNB algorithm is improved for text classification by McCallum and Nigam in 1998 (McCallum and Nigam, 1998). MNB is a type of Naive Bayes algorithm that calculates the similarity between documents based on the frequency of words occurring in a document, regardless of the order of words in a document (Kim et al., 2002). MNB performs better than the Multivariate Bernoulli model since it takes into account frequency information, that is, number of word occurrences in a document (Rennie et al., 2003). The Multinomial model serves to achieve classification better than the Multivariate Bernoulli model when working with a large vocabulary size (Raschka, 2014). The MNB calculation is shown in Equation 25.

$$P(x/c_i) = \prod_{k=1}^{n} P(w_k/c_i) \quad (25)$$

where x is a vector that include word numbers belonging each sample, c is a vector that includes class of each sample.

3.8.3. Complement Naive Bayes (CNB) Algorithm

The Complement Naïve Bayes algorithm has improved as a complement class type of MNB from Rennie et al., (2003). CNB algorithm has developed to handle the problems caused by MNB algorithm when the number of training documents for the classes is different. In other words, CNB has emerged in order to eliminate imbalances in the training data numbers of the classes (Jiang et al., 2016). It is waited that MNB and CNB algorithm show the same classification performance

when training data are distributed balanced. The Complement Naive Bayes calculation is shown in Equation 26.

$$(x/c_i) = \prod_{k=1}^{n} \frac{1}{P(w_k/c_i)} \quad (26)$$

3.8.4. Gaussion Naive Bayes (GNB) Algorithm

Naive Bayes can be performed for continuous data as well as for categorical data (Raschka, 2014). The GNB algorithm copes with continuous data such as TF-IDF matrix (Bouckaert, 2004; Tripathy and Rath, 2017). The GNB is more convenient for continuous data since it presumes that probabilities of input features are distributed according to a Gaussian distribution, which is entitled as a Normal distribution, as shown in Equation 27. It performs the classification by calculating the words occurrence probability, across all classes of data except the focused class, as shown in Equation 28 (Komiya et al., 2011).

$$P(x/c_i) = \frac{1}{\sqrt{2\pi\sigma_{c_i}^2}} e^{\frac{-\left(x-\mu_{c_i}\right)^2}{2\sigma_{c_i}^2}} \quad (27)$$

$$P(w_k \backslash c_i) = \prod_{\bar{c}} \frac{1}{w_k \backslash \bar{c}} \quad (28)$$

where x is a vector that include word numbers belonging each sample, c is a vector that includes class of each sample, $\bar{c}$ is symbolized all classes without ci.

3.9. Support Vector Machine (SVM)

SVM, which is a supervised machine learning model, has been used as a promising algorithm to classify data and perform regression in many areas such as

bioinformatics (Chowdhary et al., 2010), natural language processing (Matic et al., 1993), and computer vision (Chowdhary et al., 2010). SVM has been used to cope with many bioinformatics problems such as prediction of structure of protein secondary and tertiary, interaction of protein-protein, and prediction of splicing site (Chowdhary et al., 2010).

Boser, Guyon, and Vapnik developed the first SVM algorithm in 1992 (Bose et al., 1992). The SVM builds a decision boundary to partition the two classes. Separating the two classes also makes it qualify as a binary classifier. It performs this classification for both linearly separable and nonlinearly separable problems. The linear decision boundary is called the hyperplane, which provides the distance between two classes as far as possible. Since the decision boundary should be adjusted to get the correct result when testing new data, it should be set to maximize the margin closest to the boundary lines of both classes. The support vectors are the points closest to both classes' boundary lines. Since the algorithm is developed depending on these support vectors, the algorithm is named Support Vector Machines.

SVM performs well when allocating classes as it has a good margin separator. While SVM works effectively on high-dimensional data; works less effectively on large amounts of data. In this case, SVM works best when there is little data and high dimesional. It also works less effectively in problems with multi-classes.

SVM improves two techniques to classify only nonlinearly separated problems in the original space. The first of them is soft margin hyperplane that has a penalty function, and the other is a kernel trick where the original space is turned into a higher dimension feature space (Matic et al., 1993; Chowdhary et al., 2010; Papageorgiou et al., 1998; Frezza, 2013).

SVM has effective regularization methods. While implementing SVM; since the training time alters depending on the sample number and sample features, that is, the data size, it is decided whether to follow an explicit mapping or a kernel-based way depending on the size (Chang et al., 2010).

SVM increases machine learning rate by using kernel trick in solving problems of nonlinearly separable data and high-dimensional data. Kernel functions make it possible to solve a nonlinear problem using a linear classifier. The kernel function in SVM typically moves high-dimensional data in low-dimensional to high-dimensional space to solve it correctly. There are many varieties of kernel functions, such as sigmoid, polynomial, radial basis function (RBF), and linear. Choosing an effective kernel function is quite a difficult task.

3.10. K-Nearest Neighbors (KNN) Algorithm

Fix, E. and Hodges, J.L is proposed KNN algorithm in 1951 (Fix and Hodges, 1951). KNN is a sample-based learning model that is so called memory-based learning model. Sampled-based methods, on the other hand, only compute on all samples; they do not calculate on a specific part or summary of the samples. They are also called memory-based learning because they create a very serious need for memory. One of the sampled-based models problems is this memory requirement. Hence, the KNN method has this problem despite its success. This learning type is also named lazy learning. Because such learning is independent on the parameters, when a new sample comes, they make the computation. KNN's non-parametric model fetature provides greater modeling flexibility depending on the samples. The KNN method successes in many different problems (Kumar et al., 2005).

Since KNN performs data generation processes on infinite possibilities instead of explicitly modeling, it is ordinarily assumed as a discriminative model.

KNN is a preferred supervised learning algorithm in problems because of its easy implementation. However, its good results in noisy training data increase its popularity.

When it is considered that the dataset's sample number is n, KNN sets each sample as a point in n-dimensional space. The KNN algorithm computes each sample distance into all samples, that is, from n samples in the n-dimensional dataset. The neighbors of a sample in the KNN algorithm are k samples with the least distance to the sample among all samples. The KNN algorithm performs classification with the idea that each sample contains a class similar to its nearest neighbors. Hence, KNN decides the new sample's class label based on the k nearest neighbor samples. The majority class is appointed to each new instance depending on its nearest neighbors' class (Fix and Hodges, 1951; Kumar et al., 2005). (Fix and Hodges, 1951; Kumar et al., 2005).

IN KNN algorithm, linear time complexity arises depending on the sample size. KNN's distance calculation of all training samples induces a major computational cost. This major cost also ensues in the conclusion which this version of the KNN method is useless for big data. Extended versions of KNN algorithms have been built to minimize time complexity (Deng et al., 2016).

3.11. Deep Learning

In traditional programming, both the problem data and solution steps are given to the computer to cope with the problem. Expressly, the problem resolving steps with problem data are given into the computer. Unlike traditional programming, only the data entry is provided to the software when resolving the problem utilitizing in artificial neural networks. Therefore, deep neural network automatically extracts a solution from the data (Nielsen, 2015).

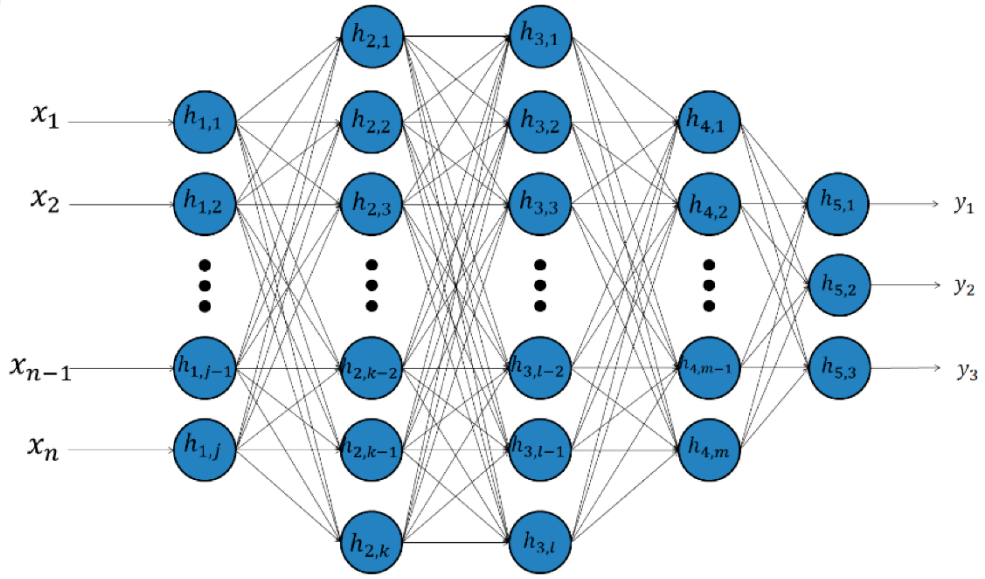

Figure 3.2. Deep neural network architecture (Berman et al., 2019)

The deep learning algorithm realizes learning from both data and data representation using many nonlinear sequential layers. Each layer utilizes the previous layer's output as input. The output emerges by processing features coming to a layer. This layer's output finds certain features at the layer level. The features it produces as output have served as following layer's input. By this, the deep learning algorithm has performed to take the low-level features. The low-level features are

served as input to the sequential layer to generate new features. The features are also created from this entry in this layer. The algorithm continues thus until the last layer, as shown in Figure 3.2. In such wise that one layer's output is the following layer's input, the deep learning algorithm resolves from the data representation (Nielsen, 2015; Tim, 2015; Guo et al., 2016).

3.11.1. Software Packages and Programmıng Languages Used for Deep Learning

There are many different software packages developed for deep learning. The most widely used of these different software packages and with which programming language the software packages will be used is shown in Table 3.2. Caffe (http://caffe.berkeleyvision.org/) is a deep learning library designed in Python to solve computer vision problems. Caffe's main benefit comes from containing ready-to-use trained networks. TensorFlow is a widely used deep learning library with open source code developed in the Python language that uses data stream graphics with flexible architectural features that can run on both GPUs and CPUs. Theano is a research platform that allows you to create your own deep learning neural network classes, developed in the Python language, that can run on both GPU and CPU. Torch is a deep learning framework that is frequently used in image processing, aiming to simplify the process that can be used in CUDA, C, and C++, which can also work on the GPU.

Table 3.2. Software Packages that enables Deep Learning algorithms to be implemented and languages used for the Packages

	Command line	C	C++	Clojure	Java	Lua	LuaJIT	Matlab	Python	Scala
Caffe	-	-	+	-	-	-	-	+	+	-
CNTK	+	-	-	-	-	-	-	-	-	-
Deeplearning 4jK	-	-	-	+	+	-	-	-	-	+
Wolfram Math.	-	-	+	-	+	-	-	-	-	-
Tensorflow	-	-	-	-	-	-	-	-	+	-
Theano	-	-	-	-	-	-	-	-	+	-
Torch	-	-	-	-	-	+	+	-	-	-
Keras	-	-	-	-	-	-	-	-	+	-
Neon	-	-	-	-	-	-	-	-	+	-

3.11.2. Convolutional Neural Networks (CNN)

While designing the CNN method, the living things visual cortex neurobiological model has inspired (Gu, et al., 2018). It was discovered that the animal visual cortex cells are responsible for sensing light by Hubel and Wiesel in 1959 (Hubel and Wiesel, 1968). Kunihiko Fukushima introduced the neocognitron in 1980 (Fukushima and Miyake, 1982). LeCun et al. introduced the CNN method on the basis of the neocognitron (LeCun, 1989).

The name of the CNN term comes from its convolution layer. The CNN method's convolution layer has been utilized as the feature extraction layer. The CNN method similarity to traditional artificial neural networks is in the training method of the features. On CNN, features are trained to optimize themselves. The main alteration between CNN and traditional ANN is the part of obtaining features (O'Shea and Nash, 2015). CNN consists of input, hidden, and output layer. In CNN, features are extracted through the convolution process from the data received through the input layer. The convolution process in the convolution layer is performed with the specified filter. In the input matrix, the feature vector is formed by traversing the entire matrix through the filter. The ReLU function is used after the convolution layer. The pooling step is performed after each convolution step.

The pooling is done in two techniques: one of them is maximum pooling and average pooling. After this feature extraction and size reduction, CNN performs like a traditional multilayer neural network (LeCun, 1989; O'Shea, and Nash, 2015).

LeCun (1989), introduced DCNN (Deep Convo) network that is performed using linear convolutions and nonlinearities on more than five layers. DCNN is composed of many neural network layers. DCNN ensures extraordinary regression and classification outcomes for high dimension. DCNN propagation is separated into training and inference with binary weights and activations. DCNN succeeds excellently if the data, computing and storage resources increases (Dos Santos and Gatti, 2014; Hochreiter and Schmidhuber, 1997; Mallat, 2016).

3.11.3. Recurrent Neural Networks (RNN)

All inputs and outputs that create a training neural network from inputs are not dependent on each other in a general neural network. The features of neural networks lead to not getting the desired result for each problem. Unlike such a neural network, RNN is a deep learning architecture type based on the input and outputs that the neural network obtains are dependent. Hence, the RNN architecture's idea is related inputs and outputs to each other. RNN is based on enabling to learn the temporal change of sequential data. An RNN is a neural network which contains loop connection.

RNN is a deep learning algorithm which takes one layer of results back into the same layer. Generally, neural networks restrict only the same fixed-length vectors as input and output to a one-to-one relationship, while the RNN architecture allows the data to be created in 5 different ways using multiple vectors, not a vector containing features. These:

a. One to One RNN
b. One to Many RNN
c. Many to One RNN
d. Many to Many RNN
e. Bidirectional Many-to-Many

The main logic of a deep neural network is that it consists of numerous hidden layers. The main logic of a recurrent neural network is that it has a hidden layer's recurrent connection. Each hidden unit is linked to both itself and all other nodes in the hidden layer. Therefore, the main logic of DRNN is that it has numerous recurrent hidden layers (Hermans and Schrauwen, 2013).

RNN overcomes training to train sequential data, such as biological text data. Sequential data comprehends correlations between close data points in the sequence, such as DNA sequences and protein sequences etc. (Schuster and Paliwal, 1997).

3.11.4. Long Short Term Memory Networks (LSTM)

In 1997, Hochreiter and Schmidhuber presented the LSTM (Hochreiter and Schmidhuber, 1997). These networks are preferred over the RNN architecture in terms of long time-lapse dependencies. Because LSTM has shown more successful results in operations performed at long time intervals than the RNN architecture. Since discovering of the information storage with recurrent backpropagation takes long time, the gradient-based LSTM method has created. Hence, these networks solve complicated and unsolved problems using RNN algorithms. These networks contain memory blocks that make them more successful than the RNN architecture in any hidden layer. Each memory block consists of memory cells and special multiplication units. Memory cells are self-connected cells that accumulate the neural network temporal state. Special impact units are gates that control the

information flow. This is an input gate to transmit input data, an output gate to transmit output data, the Sigmoid gate, which is the gate to be taken from the output gate and finalized data with the Sigmoid function, and the forget gate which is not included in the RNN algorithm but decides the amount of forgotten data from the previous cell.

LSTM is modeled more accurately than RNN in terms of long-term dependencies (Hochreiter and Schmidhuber, 1997). Since discovering to store information on long time intervals by recurrent backpropagation takes a long time, LSTM, which is a gradient-based method, is modeled. Therefore, LSTM solves complex and unsolved tasks using RRN algorithms (Hochreiter and Schmidhuber, 1997). LSTM's computational complexity is O (1).

The LSTM comprises memory blocks that make it more successful than RNN in any hidden layer. Each memory block is comprised of memory cells and special multiplicative units. Memory cells are cells with self-connections accumulating the neural network's temporal state. Special multiplicative units are gates that check information flow. These are an input, an output, and the forget gate (Sak et al., 2014).

3.11.5. Restricted Boltzmann Machines (RBM)

Boltzmann machines (BMs) are bidirectionally connected stochastic recurrent neural networks. BM is built to solve unknown significant probability distribution aspects in samples. Since this learning process is hard and time consuming, a new neural network called Restricted Boltzman Machines (RBM) has been designed by introducing restrictions to the network topology of Boltzman Machines (Hinton and Sejnowski, 1983).

An RBM is a probabilistic neural network. The RBM is called bipartite because it consists of two layers. The first layer is the input from which the data features are taken to visible nodes. The second is the hidden layer. The nodes' learning process takes place in the hidden layer. Each feature of all the data is taken to a different visible node in the input layer in the algorithm. In RBM, the visible nodes can receive input and output from outside but cannot connect among themselves. On the contrary, hidden nodes are interrelated with each other. The hidden nodes are not input and output from the outside. Each feature information from visible nodes is transmitted to a node in the hidden layer and computed for learning. The results of the calculation are processed through an activation function. RBM performs the calculation using the Contrastive Divergence method for learning purposes. In RBMs, an unequal and desired number of hidden layers, input, and output nodes can be processed.

3.11.6. Deep Belief Networks (DBN)

DBN is a probabilistic generative model. DBN provides hierarchical learning that combines RBM layers with bidirectional communication but without communication between nodes of the layers. DBN architecture is trained sequentially by sequencing each RBM layer which is the hidden layer as the visible layer of the next RBM layer. Each feature RBM layer captures strong high-order correlations among features in the layer. The DBN architecture adds a softmax activation function in the last layer after the RBM layers when used for data classification or clustering purposes.

3.11.7. Deep Auto-Controller (DAE)

DAE is a deep learning architecture that contains a feed forward multilayer neural network. The DAE is used for unsupervised learning that aims to decrease data dimension. The DAE structure has three layers as input, hidden, and output. DAE is a method that purposes to find a function that will give an output from the data taken as input. Hence, the number of neurons on the input and output layer is equal. The extraction of neuron numbers is not on the hidden layer. In DAE, the most appropriate number of neurons is selected to learn the best features for the representation of the input and to maximize performance in the hidden layer.

3.11.8. Applications of Deep Learning

Since the deep learning architecture has immense developed today, it performs better than traditional programming in the severe areas such as computer vision, speech recognition, natural language processing, bioinformatics and biomedical data processing. Deep learning architecture has performed in many diseases diagnosis, such as diagnosis of Parkinson's disease, knee calcification diagnosis, medical imaging, breast cancer, cancer diagnosis, melanoma skin cancer diagnosis, emphysema diagnosis.

4. EXPERIMENTAL RESULTS AND DISCUSSION

In the thesis, remote homology of obtained proteins from the SCOP protein database is detected by coding with the Python programming language with Keras and Biopython libraries in Anaconda environment.

4.1. Confusion Matrix

A confusion matrix gives information about predicted classes according to actual classes. The matrix is basically two-dimensional as actual and predicted classes. The confusion matrix is n×n dimensional matrix in which n is the number of classes of the outcome or class variable in the dataset. The confusion matrix in Table 4.1 is used to assess the performance of binary classification problems. In this study, the minority class is labelled as positive while the majority class is labelled as negative.

Table 4.1. The Confusion matrix on binary classification

	Predicted Positive	***Predicted Negative***
Actual Positive	*TP*	*FN*
Actual Negative	*FP*	*TN*

4.2. Evaluation Metrics

TP, TN, FP, and FN are represented as True Positive, True Negative, False Positive, and False Negative, respectively. The accuracy and ROC score are based on these TP, FP, TN, and FN parameters. The accuracy and ROC score are test success evaluation metrics. There is their explanation in detail in the subsections.

4.2.1. Accuracy, Sensitivity and Specificity

Classification accuracy, as shown in Equation 31, is an important success metric that depends on specificity, sensitivity, and predictive power (Gouvier et al., 1998; Hennekens and Buring, 1987). While specificity is the true negative rate as shown in Equation 30; sensitivity is the true positive rate in a testing process as shown in Equation 29.

$$Sensitivity = TPR = \frac{TP}{TP+FN} \quad (29)$$

$$Specificity = TNR = \frac{TN}{TN+FP} \quad (30)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (31)$$

4.2.2. AUROC Score anc ROC Curve

The ROC is a curve by positioning the TPR over the FPR rate to demonstrate the task classifier performance. Each point in the ROC space represents a different classifier performance (Bradley, 1997). ROC is a very useful test metric for testing classifier success, especially for the problems with unbalanced data, because ROC provides a visual representation of correct and incorrect predictions. The ROC is drawn with sensitivity-specificity values that depend on the problem-specific threshold value.

4.3. Obtaining Feature Vectors from Protein Sequences

In the thesis, samples are protein sequences that basically each consist of 20 different amino acids. Based on the knowledge of the 20 amino acids, it can be remarked that protein sequences have their own alphabet. Since proteins are constituted by a combination of the amino acids in different numbers and orders, can be shown with single-letter codes of the amino acid. Hence, the studies on protein sequences resemble natural language processing applications. In the thesis, natural language processing methods such as n-grams, bag of words models, TF-IDF weighting are used.

4.3.1. n-grams

n-gram slices have extracted from protein sequences. In this thesis, the experiments have carried out by subtracting different n-grams between 2 grams and 9 grams. Trigram residues of protein sequences are as listed in Table 4.2.

4.3.2. Bag of Words Model

In the study, it is presumed that protein sequences have considered as a document, n-grams of various lengths consisting of amino acids are also considered as words or terms. Bag is created with different words in all proteins of the dataset in the study. This bag can also be called a dictionary (El-Din, 2016).

The trigram-protein sequence matrix is based on the assumption of each trigram as a word and the assumption of each protein sequence as a document is shown in the Table 4.3. and Table 4.4. The trigram-protein sequence matrices in Table 4.3 and Table 4.4 are two-dimensional vectors with M rows and N columns. M is number of all words that are all trigrams and N is samples number. The samples are protein sequences in the thesis. While each entry (M, N) represents the occurence

of term M in document N in Table 4.3, each entry (M, N) represents the frequency of term M in document N in Table 4.4.

Table 4.2. Protein trigram samples

Name, Family and Superfamily of Protein	d1bsha1 1.2.7.1.1 (87-138) Epsilon subunit of F1F0-ATP synthase C-terminal domain {Escherichia coli}
Protein Sequence	QDLDEARAMEAKRKAEEHISSSHGDVDYAQASAEL AKAIAQLRVIELTKKAM
Trigram	QDL
Trigram	DLD
Trigram	LDE
Trigram	DEA
Trigram	EAR
Trigram	ARA
Trigram	RAM
Trigram	AME
Trigram	MEA
Trigram	EAK
Trigram	…
Trigram	KAM

Protein classification is performed using the bag of words matrices with both occurences and frequency. Although the bag of words model or word-documet matrix is a useful technique to obtain a fixed-length feature vector; the fact that word orders are not taken into account is a factor that may negatively affects the performance.

Table 4.3 Binary bag of words matrix of protein

	P1	P2	P3	…	PN-2	PN-1	PN
T1	1	0	1	…	0	0	0
T2	0	0	0	…	1	0	0
T3	0	0	0	…	0	0	0
…				…			
TM-2	0	0	0	…	0	0	1
TM-1	0	0	0	…	0	0	0
TM	0	0	0	…	0	0	0

Table 4.4 Protein word-document matrix

	P1	P2	P3	…	PN-2	PN-1	PN
T1	1	0	2	…	0	0	0
T2	0	0	0	…	1	0	0
T3	0	0	0	…	0	0	0
…				…			
TM-2	0	0	0	…	0	0	1
TM-1	0	0	0	…	0	0	0
TM	0	0	0	…	0	0	0

4.3.3. TF-IDF Weighting

Although the bag of words model holds the information that the term in a document does not occur or how many times it is; is not fully sufficient to express the importance of the term for the document. TF-IDF weighting is used in this thesis because it is a measurement method that is frequently used in natural language processing problems to convert text dataset to numerical expressions and shows the importance of each term.

Table 4.5 Term frequency matrix based on trigrams of protein sequence

	P1	P2	P3	…	PN-2	PN-1	PN
T1	0.00699301	0.	0.0137931	…	0.	0.	0.
T2	0.	0.	0.	…	0.00847458	0.	0.
T3	0.	0.	0.	…	0.	0.	0.
…	...	…		…	…		
TM-2	0.	0.	0.	…	0.	0.	0.01010101
TM-1	0.	0.	0.	…	0.	0.	0.
TM	0.	0.	0.	…	0.	0.	0.

Table 4.6. Inverse document frequency based on trigrams of protein sequence

T1	2.58537718
T2	4.11571091
T3	3.08007342
…	…
TM-2	4.54974939
TM-1	5.81344143
TM	5.08255392

Table 4.7. TF-IDF matrix based on trigrams of protein sequence

	P1	P2	P3	…	PN-2	PN-1	PN
T1	0.01808	0.	0.03566	...	0.	0.	0.
T2	0.	0.	0.	…	0.03488	0.	0.
T3	0.	0.	0.	…	0.	0.	0.
…	...			...			
TM-2	0.	0.	0.	…	0.	0.	0.04596
TM-1	0.	0.	0.		0.	0.	0.
TM	0.	0.	0.		0.	0.	0.

4.4. Cosine Similarity

It has been applied to use the word-document matrix rows obtained by using trigrams of protein sequence samples as samples representing proteins. This samples are used to classify remote homolog and non-remote homologous proteins using cosine similarity In Table 4.8, while P stands for protein; the numbers from 1 to N also represent the number of protein samples. For the n-length protein sample

dataset, the cosine similarities from P1 to PN are as shown in Table 4.8. However, cosine similarity on word document matrix has shown which is not successful enough in remote homology detection problems. Then, a cosine similarity method set on TF-IDF weighting has been also applied for remote homology problem. For the n-length protein sample dataset, the cosine similarities based on TF-IDF weighting from P1 to PN are as shown in Table 4.9. However, cosine similarity set on TF-IDF weighting also has shown which is not successful enough in remote homology detection problems. Although the accuracy values are high, it is noticed that the principal problem is caused by unbalanced data. Therefore, smoothing operations are performed after TF-IDF weighting. Cosine similarity results of protein samples based on TF-smothIDF weighting are as shown in Table 4.10.

Table 4.8. Cosine similarity between proteins based on word document matrix

	P1	**P2**	**P3**	**..**	**PN-2**	**PN-1**	**PN**
P1	1.	0.06851	0.07812	..	0.01540	0.04181	0.01664
P2	0.06851	1.	0.14031	..	0.04525	0.01365	0.03261
P3	0.07812	0.14031	1.	..	0.03583	0.01946	0.00775
…	…			..			
PN-2	0.01540	0.04525	0.03583	..	1.	0.02301	0.00916
PN-1	0.04181	0.01365	0.01946	..	0.02301	1.	0.03317
PN	0.01664	0.03261	0.00775	..	0.00916	0.03317	1.

Table 4.9. Cosine Similarity between proteins based on TF-IDF weighting

	P1	P2	P3	…	PN-2	PN-1	PN
P1	1.	0.05395	0.05589	…	0.01208	0.03057	0.01105 6
P2	0.05395	1.	0.09762		0.03428	0.00813	0.02135
P3	0.05589	0.09762	1.	...	0.03286	0.01333	0.00408
…	...			...			
PN-2	0.01208	0.03428	0.03286	...	1.	0.02066	0.00810
PN-1	0.03057	0.00813	0.01333	...	0.02066	1.	0.03077
PN	0.01106	0.02136	0.00408	...	0.00809	0.03077	1.

Table 4.6. Cosine similarity between proteins based on TF-smothIDF weighting

	P1	P2	P3	…	PN-2	PN-1	PN
P1	1.	0.07335	0.08538	...	0.01666	0.04595	0.01893
P2	0.07337	1.	0.15388	...	0.04926	0.01572	0.03718
P3	0.08538	0.15388	1.	...	0.03692	0.02165	0.00922
…	...			…			
PN-2	0.01666	0.04926	0.03692	...	1.	0.02387	0.00959
PN-1	0.04595	0.01572	0.02165	...	0.02741	1.	0.03424
PN	0.0189	0.03718	0.00922	…	0.00959	0.03424	1.

4.5. Protein Superfamily and Family Classfication Using n-gram

In the thesis, family and superfamilies of protein to detect remote homolog protein are found from protein sequences using Biopython tool (https://biopython.org/) on Python programming language.

Remote homologous proteins are based on small sequence similarity. They can also define as proteins from the same superfamily and different families. Hence, proteins with different n-grams have classfied into superfamilies and families utilizing severe machine learning algorithms. There strores extracted n-grams from the protein data in NoSQL database.

The various methods such as SVM, Naive Bayes, deep learning, and k-nearest-neighbor methods have applied successfully for various problems such as detection of remote homolog proteins, illness recognition, and text classification

(Leslie et al., 2001; McCallum et al., 1998; Şeker et al., 2017; Liu et al., 2014). The various methods such as Naive Bayes, SVM linear, SVM scale, deep learning, and k-nearest-neighbor methods have been applied to classify into superfamilies with different n-grams in the thesis. The algorithms have given between 20% and 100% of accuracy, as shown in Table 4.11. Similarly, the methods are used to classify into families with different n-grams between 0.6 and 1 accuracy, as shown in Table 4.12.

Table 4.7. Protein superfamily classification accuracy based on n-gram

Superfamilies	ngr= 2,4	ngr= 3,4	ngr= 4,4	ngr= 2,5	ngr= 2,2	ngr= 3,3
Naive Bayes	0.33-1.0	0.6	1	0.2	0.2	0.3
SVM linear	0.33-1.0	0.6	1	0.2-0.4	0.2	0.3
SVM scale	0.33-0.66	0.6	1	0.2-0.4	0.2	0.3
Deep learning	0.66-1.0	0.8	0.33	0.4-0.6	0.2	0.3
k-nearest-neigbor	0.6-1.0	0.8	1	0.8	0.2	0.3

Table 4.8. Protein family classification accuracy based on n-gram

Families	ngr= 2,4	ngr= 3,4	ngr= 4,4	ngr= 2,5	ngr= 2,2	ngr= 3,3
Naive Bayes	0.8	0.8	0.8	0.8	1.	0.6
SVM linear	0.8	0.8	0.8	0.8	1.	0.6
SVM scale	0.8	0.8	0.8	0.8	1.	0.6
Deep learning	0.8	0.8	0.8	0.8	0.8	0.6
k-nearest-neigbor	1.	0.8	0.8	1.	1.	0.6

4.6. Naive Bayes Method

In this section, performance comparisons have been made using different Naive Bayes algorithms to explore remote homologous proteins over the features obtained using the proteins, n-gram, and TF-IDF weighting method.

The obtained results shown in Table 4.13 show a mean success of all Naive Bayes classifications using various n values for n-gram on protein classification. Performance of Multinomial Naive Bayes (MNB) algorithm on the TF-IDF matrix to determine remote homolog protein has the same performance of Bernoulli Naive Bayes (BNB) algorithm even in different n-gram trials. Similarly, the performance of the Complement Naive Bayes (CNB) algorithm on TF-IDF matrix to determine remote homolog protein has the same performance as those of the Gaussian Naive Bayes (GNB) algorithm even in different n-gram trials. There give the mean results of Mean Absolute Error (MAE) on the Naive Bayes classification of remote homologous protein in the Table 4.14 depending on different n-gram numbers. The mean results of Mean Squared Error (MSE) on the Naive Bayes classification of remote homologous protein depending on different n-gram numbers are given in Table 4.15.

Table 4.9. Mean accuracy success of the Naive Bayes classification of protein

N-GRAM	BNB	MNB	CNB	GNB
2,3	0.99796	0.99796	0.99801	0.99806
2,4	0.99803	0.99803	0.99797	0.99813
2,5	0.99797	0.99797	0.98558	0.99807
2,7	0.99391	0.99391	0.97377	0.99806
2,9	0.98813	0.98813	0.97442	0.99806

Table 4.10. Mean Absoulute Errors (MAE) of the Naive Bayes classification algorithms

Char n-grams		BNB	MNB	CNB	GNB
min n-gram length	max n-gram length				
2	3	0.00381	0.00381	0.00361	0.00360
2	4	0.00378	0.00378	0.00361	0.00361
2	5	0.00452	0.00452	0.00358	0.00358
2	7	0.01525	0.01525	0.00361	0.00361
2	9	0.02807	0.02807	0.00366	0.00366

Table 4.11. Mean Squared Errors (MSE) of the Naive Bayes classification algorithms

char n-grams		BNB	MNB	CNB	GNB
min n-gram length	max n-gram length				
2	3	0.00381	0.00381	0.00360	0.00360
2	4	0.00378	0.00378	0.00361	0.00361
2	5	0.00452	0.00452	0.00358	0.00358
2	7	0.01525	0.01525	0.00361	0.00361
2	9	0.02807	0.02807	0.00366	0.00366

Table 4.12. Execution Time of the Naive Bayes classification algorithms

Execution Time (sec)	3452 Iterations	2000 Iterations	1000 Iterations
BNB	$31x10^{-4}$	$13x10^{-4}$	$6{,}4x10^{-4}$
MNB	$26x10^{-4}$	$12x10^{-4}$	$6x10^{-4}$
CNB	$27x10^{-4}$	$13x10^{-4}$	$6{,}5x10^{-4}$
GNB	$24x10^{-4}$	$10x10^{-4}$	$5{,}4x10^{-4}$

There give execution time on the Naive Bayes classification of remote homologous proteins in Table 4.16. ROC curve of the Naive Bayes classification for remote homologous proteins between 2 and 3 grams is shown in Figure 4.1. The Naive Bayes classification's ROC curve for remote homolog protein between 2 and 4 grams is shown in Figure 4.2. The Naive Bayes classification's ROC curve for

remote homolog protein between 2 and 5 grams is shown in Figure 4.3. ROC curve of the Naive Bayes classification for remote homolog protein between 2 and 7 grams is shown in Figure 4.4. The Naive Bayes classification's ROC curve for remote homolog protein between 2 and 9 grams is shown in Figure 4.5. Although the mean accuracy success of the Naive Bayes classification of protein sequences is quite high and proposed, AUROC values are very low. According to AUROC equals 0.5, the confidence of the Naive Bayes classification of remote homologous proteins is not acceptable. While accuracy is high; the reason for the high AUROC values is the dataset because the dataset is seen as unbalanced. The unbalanced dataset problem has been demonstrated in the remote homology of protein. The dataset on the remote homology problem needs extra preprocessing to classify successful.

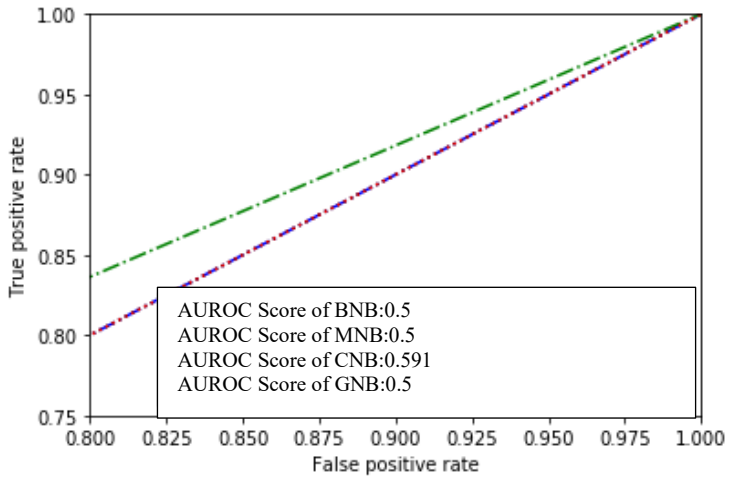

Figure 4.1. The Naive Bayes classification's ROC curve between 2 and 3 grams for d1mbk protein sequence sample

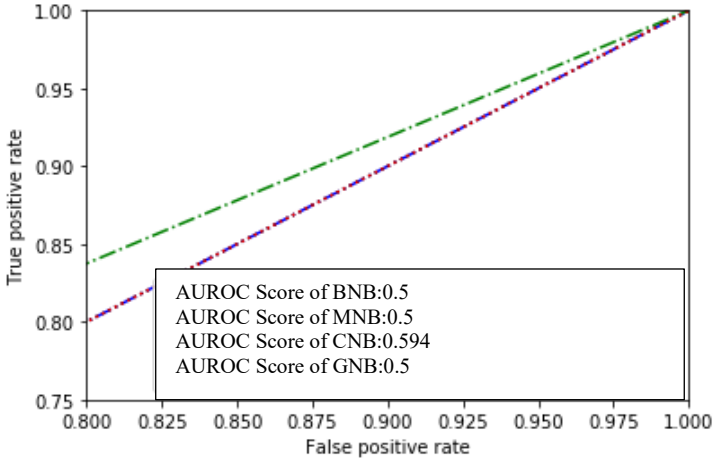

Figure 4.2. The Naive Bayes classification's ROC curve between 2 and 4 grams for d1mbk protein sequence sample

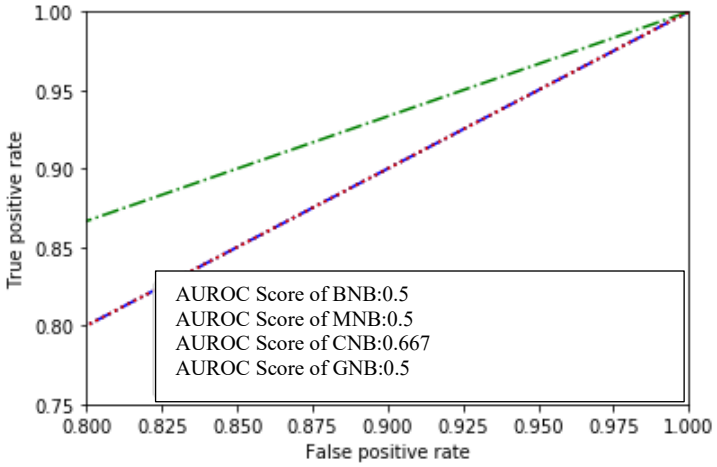

Figure 4.3. The Naive Bayes classification's ROC curve between 2 and 5 grams for d1mbk protein sequence sample

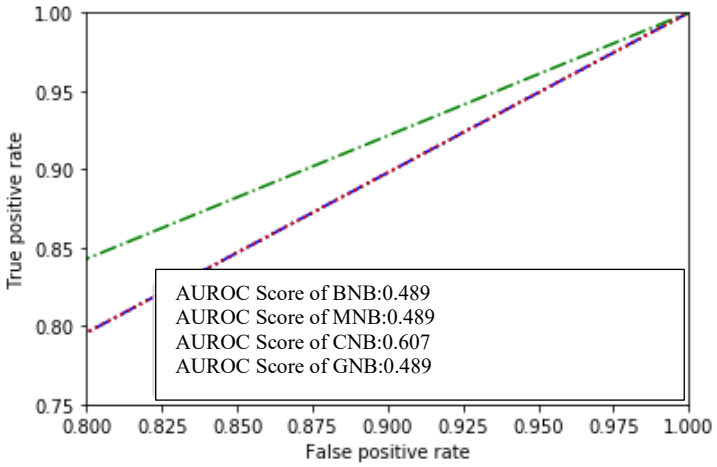

Figure 4.4. The Naive Bayes classification's ROC curve between 2 and 7 grams for d1mbk protein sequence sample

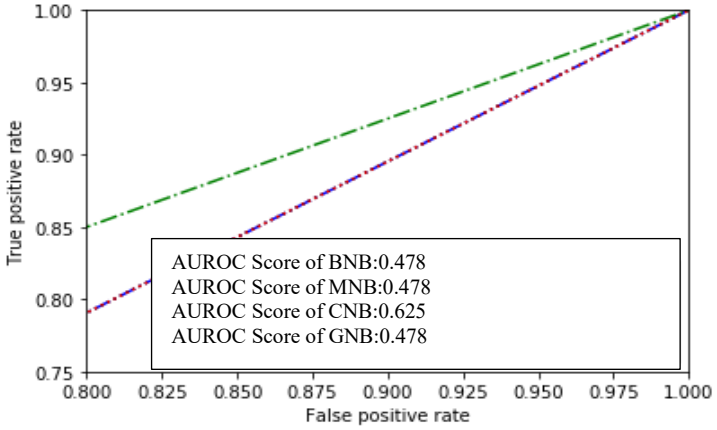

Figure 4.5. The Naive Bayes classification's ROC curve between 2 and 9 grams for d1mbk Protein Sequence Instance

4.7. Deep Learning Method with Smoothing Features

In the thesis, the deep learning method has been coded with Python programming language with Keras backend Tensorflow, and Biopython libraries. The thesis first step is to obtain a protein dataset. The second is to explore the family and superfamily of proteins on the dataset. There have determined remote homologous proteins in the third step. Identifying homolog and remote homologous proteins with each other have determined the proteins' sequences' similarity. A homolog definition is that proteins share a sequence similarity of more than 40%. Moreover, proteins having a pairwise sequence similarity between 20% and 40% are defined as remote homologs. The identifying of whether the proteins are homolog and remote homolog to each other can be decided by recognizing the families and superfamily of the proteins. Then, the thesis basic structure has built on this knowledge. There has been established a new system that uses both sequence similarity and family and superfamily similarity in the thesis. Proteins that belong to the same superfamily and different families are remote homologous proteins with each other and in the meantime proteins which belong to the same superfamily and same family are homolog proteins with each other. Hence, in the thesis, for coding the result, 1 is used as the remote homolog protein tag and 0 is used for the non-remote homolog protein.

There has been created by labeling proteins from the same superfamily and different families as remote homologous proteins and proteins from the different folds as non-remote homologous proteins in the thesis. In our system, positive test samples are taken from proteins from within the target family. Positive train samples are taken proteins from within the same superfamily of protein with the target family and outside the target family.

Since remote homologous proteins look alike and have very small sequence similarities with one another, it is hard to discover remote homology with the entire and single amino acid sequence similarity of any protein. Hence, the fourth step is to take n-grams from protein sequences.

The next step is to extract the TF-IDF matrix using extracted bi-grams from protein sequences. Then, a smoothing process is performed to balance test and train protein data. The next step is classification of proteins into remote homolog and non-remote homolog using Deep Learning architecture. In fact, the "Deep" term in Deep Learning refers to the deep of layers in neural networks. The experiment deep learning architecture has composed from an input layer with 1200 neurons, 3 hidden layers with (600,300,100) neurons, and an output layer. After the input and hidden layers, the dropout function is performed to diminish overfitting. The softmax function has classified proteins into remote homolog or non-remote homolog in the output layer.

The AUROC score and confusion matrix results of remote homology detection are shown in Table 4.17. The results are with the smoothing and without the smoothing process for between 1 and 5 grams of target 1.4.1.1 family. As shown in the confusion matrix in the remote homology detection without smoothing in Table 1, all 23 positive instances for target 1.4.1.1 family are misclassified. Hence, remote homology detection without smoothing is unacceptable. In terms of confusion matrix, accuracy, and AUROC score, bigrams from 1 and 5 grams have performed the best in Table 1.

There have shown the experimental results of detection using Deep Learning of remote homology of proteins of preferred 54 families in Appendix A.1. Figure 4.6 shows that the AUROC scores and accuracy curves of 54 target families. The mean AUROC scores of various remote homology studies using NLP techniques such as

n-gram and Top-n-gram are shown in Table 4.19. Table 4.19 has showed that bigram choice is the most successful choice for remote homology detection operation in the n-gram process.

Table 4.13. Accuracy with or without smoothing of the Deep Learning of protein of the homeodomain family represented with 1.4.1.1 in SCOP 1.53 using deep learning on epoch (150) with max features = 9000.

Char n-gram length		**Confusion matrix without smooth**	**Confusion matrix with smooth**	**Acc. without smooth**	**Acc. with smooth**	**AUROC score without smooth**	**AUROC score with smooth**
Min	Max						
1	1	[1994, 0] [23, 0]	0.9886	0.9886	0.9603	0.6375	0.9626
2	2	[1994, 0] [23, 0]	0.9886	0.9886	0.9742	0.6375	0.9725
3	3	[1991, 3] [23, 0]	0.9871	0.9871	0.9831	0.6366	0.8898
4	4	[1994, 0] [23, 0]	0.9886	0.9886	0.9712	0.6375	0.7441
5	5	[1994, 0] [23, 0]	0.9886	0.9886	0.9886	0.6375	0.6375

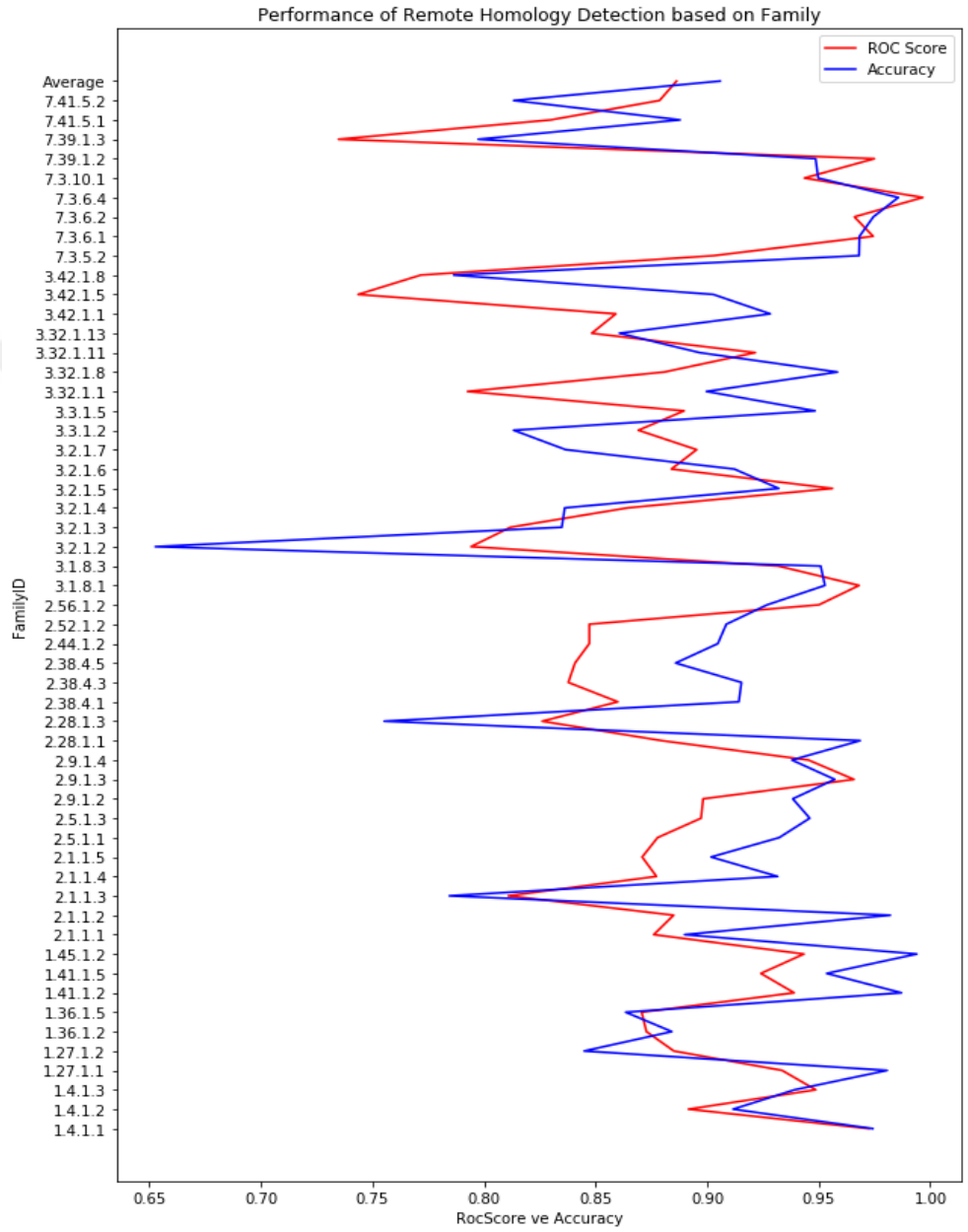

Figure 4.5. AUROC scores and accuracy results of 54 target family

Table 4.14. Mean AUROC scores reported in various remote homology studies

Methods	AUROC Score	Source
Deep learning-smoothing & TF-IDF & n-grams (n = 2)	0.8863 (mean score)	Current experiment (Scop 1.53)
Deep learning-smoothing & TF-IDF & n-grams (n = 2)	0.9967 (the best score, not the mean)	Current experiment (Scop 1.53)
Deep learning-smoothing & TF-IDF & n-grams (n = 2)	0.8690 (mean score)	Current experiment (Scop Benchmark)
Deep learning-smoothing & TF-IDF & n-grams (n = 2)	0.9937 (the best score, not the mean)	Current experiment (Scop Benchmark)
Deep learning-smoothing & TF-IDF & n-grams (n = 2)	0,8965 (mean score)	Current experiment (the new dataset from Scop207)
SVM & n-gram	0.7914 (mean score)	Dong et al., 2006
SVM & n-gram & LSA	0.8595 (mean score)	Dong et al., 2006
SVM & n-gram & p1	0.8870 (mean score)	Liu et al., 2014
SVM & n-gram & KTA	0.8920 (mean score)	Liu et al., 2014
SVM & n-grams & LDA	0.9351 (the best score, not the mean)	Yeh and Chen, 2010
SVM & TF-IDF & n-grams & LDA	0.9435 (the best score, not the mean)	Yeh and Chen, 2010
SVM & Top- n-grams (n = 1)	0.7309 (mean score)	Liu et al., 2008
SVM & Top- n-grams (n = 2)	0.7929 (mean score)	Liu et al., 2008
SVM & Top- n-grams (n = 3)	0.7740 (mean score)	Liu et al., 2008
SVM & Top-n-grams & LSA (n = 2)	0.8121 (mean score)	Liu et al., 2008

4.8. KNN Method

The results for remote homologous proteins obtained by 16 different distance method with KNN classification algorithm have shown in this section. The KNN classification algorithm has processed the features extracted from the protein sequences obtained from the SCOP 1.53 database. There are the various distance

measurement methods to calculate the distance among two pairs of samples in the most distinctive way for the problem being studied. 16 different distances in the thesis are namely Bray Curtis, Canbera, Correlation, Cosine, Dice, Chebyshev, Euclidean, Jaccard, Kulczynski, Hamming, Matching, Minkowski, SokalMichener, SokalSneath, RogersTanimoto and RussellRao distances. The accuracy results of KNN with sixteen distances for remote homology without cross validation have shown in Table 4.20.

The used dataset on remote homology causes an unbalanced data problem because the samples' number that are remote homologous with each other is low and the samples' number that are not remote homologous is high. To solve this unbalanced data problem, Stratified cross validation with a special k-fold value formula is recommended.

Class1 represents the remote homolog class, which has a low number of samples, while Class2 represents the non-remote homolog class, which has a high number of samples. They have created the formula in Equation 32. It aims to automatically discover the k number in the k-fold of the Stratified cross validation method according to the classes and theirs samples.

$$\text{k-split} = \text{test samples number of the Class1} \quad (32)$$

Accuracy results of KNN with sixteen distances for remote homology with Stratified cross validation with a special k-fold value are shown in Table 4.21. There have shown confusion matrixes of KNN with distances for remote homology without cross validation in Appendix B.1. There have shown confusion matrixes of KNN with distances for remote homology with Stratified cross validation with a special k-fold value in Appendix B.2. There have shown the precision and recall values of

KNN with distances for remote homology with Stratified cross validation with a special k-fold value in Table 4.21 and Table 4.22, respectively.

According to accuracy results and confusion matrix values, the KNN study with StratifiedKFold cross validation is seen as quite successful. But AUROC scores show that needs improvement. So that the new k-split method can be built.

The mean AUROC scores of the study with KNN have been compared with the other SVM-based methods as shown in Table 4.29. Liao and Noble (2003) claimed that SVM-based methods perform better than KNN-based methods to discover protein remote homolog proteins. This study has shown that the KNN method is also successful in detecting remote homologous proteins when the dataset is transformed into a balanced and regular dataset with the necessary preprocessing.

Table 4.15. Accuracy results of KNN with sixteen distances for remote homology without cross validation

Distance/Similarity Methods	Lowest Accuracy	Highest Accuracy	Mean Accuracy
Bray Curtis	0.95464	0.99586	0.98758
Euclidean	0.95959	0.99572	0.98715
Minkowski	0.95959	0.99572	0.98715
Dice	0.95364	0.99609	0.98736
Jaccard	0.95364	0.99609	0.98736
Chebyshev	0.95687	0.99655	0.99655
Cosine	0.94224	0.99609	0.98714
SokalSneath	0.95364	0.99609	0.98736
Correlation	0.94199	0.99609	0.98712
Matching	0.95945	0.98729	0.99609
Rogers Tanimoto	0.95945	0.99609	0.98729
Sokal Michener	0.95945	0.99609	0.98729
Canbera	0.95918	0.99609	0.98718
Hamming	0.94851	0.99609	0.98708
Kulczynski	0.95860	0.99609	0.98843
Russell Rao	0.96530	0.99609	0.98867

Table 4.20. Accuracy results of KNN with four distances for remote homology with cross validation

Distance/Similarity Methods	Lowest Accuracy	Highest Accuracy	Mean Accuracy
Bray Curtis	0.97078	0.99901	0.98968
Euclidean	0.96327	0.99951	0.98866
Minkowski	0.96327	0.99951	0.98866
Dice	0.97186	1.0	0.99039
Jaccard	0.97186	1.0	0.99039
Chebyshev	0.96735	0.99901	0.98825
Cosine	0.96717	1.0	0.98882
SokalSneath	0.97186	1.0	0.99039
Correlation	0.96717	1.0	0.98882
Matching	0.97173	1.0	0.98935
Rogers Tanimoto	0.97173	1.0	0.98935
Sokal Michener	0.97173	1.0	0.98935
Canbera	0.97000	1.0	0.98893
Hamming	0.94127	0.99803	0.98875
Kulczynski	0.96789	0.99951	0.99030
Russell Rao	0.97173	0.99951	0.99025

Table 4.16.Precision values of KNN with distances for remote homology with StrafiedKFold cross validation

Distance/Similarity Methods	1.4.1.1 Family	2.1.1.2 Family	2.28.1.1 family	3.42.1.5 family	7.3.10.1 Family
Bray Curtis	0.99588	0.99329	0.99640	0.99484	0.99232
Euclidean	0.99488	0.98718	0.99463	0.99482	0.98946
Minkowski	0.99488	0.98718	0.99463	0.99482	0.98946
Dice	0.99703	0.97462	0.99803	0.99588	0.99337
Jaccard	0.99703	0.97462	0.99803	0.99588	0.99337
Chebyshev	0.99372	0.97017	0.99442	0.99037	0.98616
Cosine	0.99745	0.97396	0.99803	0	0.99312
SokalSneath	0.99703	0.97462	0.99803	0	0.99337
Correlation	0.99797	0.97396	0.99803	0	0.99259
Matching	0.99640	0	0.99435	0	0.98986
Rogers Tanimoto	0.99640	0	0.99435	0.98948	0.98986
Sokal Michener	0.99640	0	0.99435	0.98948	0.98986
Canberra	0.99444	0	0.99411	0.98956	0.99021
Hamming	0.99507	0	0.99461	0.98214	0.99069
Kulczynski	0.99311	0	0.99838	0	0.99443
RussellRao	0.99116	0	0.99774	0	0.99417

Table 4.17. Recall values of KNN with distances for remote homology with StrafiedKFold cross validation

Distance/Similarity Methods	1.4.1.1 Family	2.1.1.2 Family	2.28.1.1 family	3.42.1.5 family	7.3.10.1 Family
Bray Curtis	0.99356	0.99231	0.99580	0.99310	0.99066
Euclidean	0.99207	0.98718	0.99352	0.99104	0.98639
Minkowski	0.99207	0.98718	0.99352	0.99104	0.98639
Dice	0.99703	0.97821	0.99806	0.99586	0.99253
Jaccard	0.99703	0.97821	0.99806	0.99586	0.99253
Chebyshev	0.99306	0.97308	0.99352	0.98828	0.98026
Cosine	0.99753	0.97692	0.99806	0.99104	0.99333
SokalSneath	0.99703	0.97821	0.99806	0.99586	0.99253
Correlation	0.99802	0.97692	0.99806	0.99586	0.99280
Matching	0.99653	0.97180	0.99255	0.99104	0.98986
Rogers Tanimoto	0.99653	0.97180	0.99255	0.99172	0.98986
Sokal Michener	0.99640	0.97180	0.99255	0.99172	0.98986
Canberra	0.99455	0.97180	0.99255	0.99172	0.99040
Hamming	0.99507	0.97180	0.99385	0.99035	0.99093
Kulczynski	0.99306	0.97180	0.99838	0.99104	0.99440
RussellRao	0.99108	0.97180	0.99773	0.99104	0.99413

Table 4.18. Precision values of KNN with distances for remote homology with k-split method

Distance/Similarity Methods	1.4.1.1 Family	2.1.1.2 Family	2.28.1.1 family	3.42.1.5 family	7.3.10.1 Family
Bray Curtis	0.87282	0.80508	0.83131	0.77941	0.95545
Euclidean	0.85171	0.84105	0.76905	0.79754	0.93080
Minkowski	0.85171	0.84105	0.76905	0.79754	0.93080
Dice	0.84065	0.75623	0.83367	0.77980	0.94413
Jaccard	0.84065	0.75623	0.83367	0.77980	0.94413
Chebyshev	0.85871	0.83131	0.72773	0.74615	0.92664
Cosine	0.82799	0.71722	0.84832	0.76658	0.93272
SokalSneath	0.84065	0.75623	0.83367	0.77980	0.94413
Correlation	0.83215	0.71785	0.85022	0.76822	0.93760
Matching	0.71147	0.66153	0.71530	0.69000	0.83603
Rogers Tanimoto	0.71147	0.66153	0.71530	0.69000	0.83603
Sokal Michener	0.71147	0.66153	0.71530	0.69000	0.83603
Canberra	0.71026	0.64068	0.71177	0.67424	0.83741
Hamming	0.58946	0.46985	0.66648	0.48273	0.79325
Kulczynski	0.49866	0.46784	0.81158	0.60713	0.78858
RussellRao	0.44980	0.43303	0.78770	0.51310	0.76226

Table 4.19. Recall values of KNN withdistances for remote homology with k-split method

Distance/Similarity Methods	1.4.1.1 Family	2.1.1.2 Family	2.28.1.1 family	3.42.1.5 family	7.3.10.1 Family
Bray Curtis	0.85701	0.79312	0.80961	0.75045	0.95879
Euclidean	0.83356	0.83129	0.74763	0.77162	0.93266
Minkowski	0.83356	0.83129	0.74763	0.77162	0.93266
Dice	0.81310	0.72661	0.80339	0.73739	0.94621
Jaccard	0.81310	0.72661	0.80339	0.73739	0.94621
Chebyshev	0.84368	0.79522	0.70108	0.72072	0.92468
Cosine	0.74299	0.66829	0.80201	0.71554	0.93169
SokalSneath	0.81310	0.72661	0.80339	0.73738	0.94621
Correlation	0.75149	0.67415	0.80454	0.71892	0.93750
Matching	0.63379	0.60483	0.70892	0.61329	0.79357
Rogers Tanimoto	0.63379	0.60483	0.70892	0.61329	0.79357
Sokal Michener	0.63379	0.60483	0.70892	0.61329	0.79357
Canberra	0.62851	0.57602	0.70154	0.60225	0.79357
Hamming	0.53770	0.50412	0.63565	0.51532	0.70238
Kulczynski	0.59563	0.54323	0.74325	0.60856	0.64698
RussellRao	0.58368	0.53152	0.68057	0.58221	0.57151

Table 4.20. AUROC scores of KNN with distances for remote homology with StratifiedKFold cross validation

Distance/Similarity Methods	1.4.1.1 Family	2.1.1.2 Family	2.28.1.1 family	3.42.1.5 family	7.3.10.1 Family
Bray Curtis	0.99492	0.94849	0.98275	0.89444	0.97783
Euclidean	0.97985	0.88996	0.96645	0.96815	0.96504
Minkowski	0.97985	0.88996	0.96645	0.96815	0.96504
Dice	0.95635	0.75900	0.94705	0.79594	0.97579
Jaccard	0.95635	0.75900	0.94705	0.79594	0.97579
Chebyshev	0.92576	0.77339	0.94436	0.84262	0.95065
Cosine	0.94281	0.71702	0.95438	0.61539	0.94090
SokalSneath	0.95635	0.75900	0.94705	0.79594	0.97579
Correlation	0.95696	0.71702	0.95438	0.79594	0.94053
Matching	0.91448	0.64350	0.97277	0.61539	0.93194
Rogers Tanimoto	0.91448	0.64350	0.97277	0.66768	0.93194
Sokal Michener	0.91448	0.64350	0.97277	0.66768	0.93194
Canberra	0.91376	0.64350	0.95028	0.69295	0.92883
Hamming	0.85696	0.64350	0.94444	0.61538	0.92939
Kulczynski	0.79883	0.64350	0.94720	0.61539	0.92815
RussellRao	0.74018	0.64350	0.93221	0.61539	0.92473

Table 4.21. AUROC scores of KNN withdistances for remote homology with k-split method

Distance/Similarity Methods	1.4.1.1 Family	2.1.1.2 Family	2.28.1.1 family	3.42.1.5 family	7.3.10.1 Family
Bray Curtis	0.91601	0.90255	0.91399	0.87135	0.98380
Euclidean	0.91084	0.92861	0.88427	0.89543	0.98133
Minkowski	0.91084	0.92861	0.88427	0.89543	0.98133
Dice	0.89475	0.86345	0.92864	0.88191	0.98384
Jaccard	0.89475	0.86345	0.92864	0.88191	0.98384
Chebyshev	0.93337	0.96723	0.86060	0.85833	0.98906
Cosine	0.93484	0.90664	0.94791	0.89194	0.97905
SokalSneath	0.89475	0.86345	0.92864	0.88191	0.98384
Correlation	0.93617	0.90344	0.94904	0.89113	0.98300
Matching	0.88064	0.85618	0.83185	0.83601	0.94934
Rogers Tanimoto	0.88064	0.85618	0.83185	0.83601	0.94934
Sokal Michener	0.88064	0.85618	0.83185	0.83601	0.94934
Canberra	0.88232	0.85901	0.83404	0.83953	0.94911
Hamming	0.86396	0.86550	0.86167	0.83417	0.94358
Kulczynski	0.92077	0.92306	0.94656	0.88212	0.97496
RussellRao	0.91667	0.92960	0.94941	0.88086	0.96984

Table 4.22. AUROC scores of KNN with distances for remote homology with StratifiedKFold cross validation

Distance/Similarity Methods	Lowest AUROC	Highest AUROC	Mean AUROC
Bray Curtis	0.52855	0.99940	0.77673
Euclidean	0.52724	1.0	0.76428
Minkowski	0.52724	1.0	0.76428
Dice	0.52874	1.0	0.72237
Jaccard	0.52874	1.0	0.72237
Chebyshev	0.52855	0.98751	0.77173
Cosine	0.52855	1.0	0.69526
SokalSneath	0.52874	1.0	0.72237
Correlation	0.52855	1.0	0.69551
Matching	0.52855	1.0	0.69007
Rogers Tanimoto	0.52855	1.0	0.69007
Sokal Michener	0.52854	1.0	0.69007
Canberra	0.52855	1.0	0.68911
Hamming	0.52855	0.99774	0.66834
Kulczynski	0.52855	0.97441	0.65961
RussellRao	0.52855	0.96517	0.64057

Table 4.23. AUROC scores of KNN withdistances for remote homology with k-split method

Distance/Similarity Methods	Lowest AUROC	Highest AUROC	Mean AUROC
Bray Curtis	0.81516	0.98921	0,92024
Euclidean	0.81795	0.98392	0.91570
Minkowski	0.79848	0.98392	0.91570
Dice	0.74609	0.98861	0,89754
Jaccard	0.74609	0.98861	0,89754
Chebyshev	0.82378	0.98906	0.91341
Cosine	0.75831	0.97920	0.91016
SokalSneath	0.74609	0.98861	0,89754
Correlation	0.76938	0.98300	0.91087
Matching	0.72844	0.97979	0.86811
Rogers Tanimoto	0.72844	0.97979	0.86811
Sokal Michener	0.72844	0.97979	0.86811
Canberra	0.72729	0.97099	0.86359
Hamming	0.69662	0.94358	0.84520
Kulczynski	0.72462	0.97632	0.88270
RussellRao	0.72516	0.99466	0.87927

Table 4.24. AUROC scores of KNN with distances for remote homology with k-split method

Distance/Similarity Methods	**Mean AUROC**	**Ref**
KNN with Bray Curtis with StratifiedKFold cross validation with n-gram	0.77673	The Proposed Study
KNN with Bray Curtis with k-split method with n-gram	0,92024	The Proposed Study
SVM-Ngram	0,81200	Lovato et al., 2016
SVM with Top Ngram	0,71720	Lovato et al., 2016
SVM-Ngram-p1	0, 88700	Lovato et al., 2016
SVM-Ngram-KTA	0, 89200	Lovato et al., 2016
VBKC	0.92400	Damoulas and Girolami, 2008
SVM (SW)	0.89600	Damoulas and Girolami, 2008
SVM (LA)	0.92500	Damoulas and Girolami, 2008
SVM (MM)	0.87200	Damoulas and Girolami, 2008
SVM (Mono)	0.91900	Damoulas and Girolami, 2008
SVM pairwise (SVM PW)	0.7329	Nakshathram et al., 2021
GPkernal	0.76210	Nakshathram et al., 2021
LSTM	0.80240	Nakshathram et al., 2021
SOFM-Top	0.82100	Nakshathram et al., 2021
SOFM-SW	0.92100	Nakshathram et al., 2021
SOFM-SMSW	0.94100	Nakshathram et al., 2021

5. CONCLUSION

Due to the increase in protein amounts, protein classification problems have become essential in molecular biology. Many studies have been carried out on homologous protein and remote homologous proteins to detect the protein families. They are among protein classification studies. Homologous proteins are among proteins with more than 40% similarity sequence identity; remote homologous proteins are proteins with sequence similarity between 20% and 40%. The lower similarity makes the remote homologous protein detection a harder problem than the homologous proteins's detection.

In this thesis, machine learning algorithms such as Naive Bayes and KNN have been applied to detect the classes of remote homologous proteins. The remote homologous proteins' detection implicitly supplies great benefit to the medical field because the structure of unknown proteins is useful in the new drugs discovery in medicine.

In this thesis, the SCOP 1.53 database, which is the database frequently used in protein classification studies, the SCOP benchmark database, and the newly created SCOP protein database from SCOP 2.07 have been used to test the thesis model. The SCOP database usage of both positive and negative samples in classification problems such as the remote homologous proteins' detection is a significant factor that increases the success of classification.

Although there are many protein alignment-based methods on protein sequences for feature extraction; protein sequences bring to mind natural language processing methods based on text and characters. To classify protein sequences with amino acids as remote homologous and non-remote homologous by machine learning algorithms, it is requisite to extract fixed-length datasets. For this objective,

bag of words model, word-document matrix and n-gram methods have used. Thus, this thesis shows that NLP techniques can be beneficially utilized in bioinformatics studies. When performing the n-gram method, the selected n value's effects on the success has showed. The features obtained by the n-gram method has not be sufficient for classification as such. Then there have obtained the features by applying the TF-IDF vectorization method.

In thesis, remote homologous protein detection has been performed using 4 different Naive Bayes classification algorithms, namely Multinomial, Complement, Bernoulli, and Gaussian Naive Bayes. The Gaussian Naive Bayes method stands out among these four methods for the remote homologous proteins detection in the success ranking. Thus, the use of various Naive Bayes algorithms in bioinformatics and their successful performance in text processing are revealed.

There have obtained the protein dataset by smoothing the protein samples in this thesis. Remote homologous proteins have been discovered by obtaining the protein dataset with the newly constructed deep learning method. The protein dataset success with the new method has an average accuracy of 89.13% and an average ROC score of 88.39%.

Another method to cope with the unbalanced data problem in this thesis has been proposed. The Strafied cross validation method with a special k-fold value formula has been proposed to diminish the unbalanced data problem on remote homologous protein detection. The KNN algorithm has shown the highest performance with 99% accuracy with Stratified cross validation with special k-fold value and Bray Curtis distance.

REFERENCES

Aggarwal, D., and Roopam, S., 2016. Emerging Technologies For Big Data Processing: NOSQL and NEWSQL Data Stores. International Journal Of Engineering and Computer Science, 5(1):15598-15604.

Al-Anzi, F. S., and AbuZeina, D., 2017. Toward an Enhanced Arabic Text Classification Using Cosine Similarity and Latent Semantic Indexing. Journal of King Saud University-Computer and Information Sciences, 29(2): 189-195.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J., 1990. Basic Local Alignment Search Tool. Journal of Molecular Biology, 215(3): 403-410.

Al-Hassai, F. A., and Kalyankar, N. V., 2013. The Classification Accuracy of Multiple-metric Learning Algorithm on Multi-sensor Fusion. Internat. J. Soft Comput. Engng, 3(4):124-131.

Andreeva, A., Howorth, D., Brenner, S. E., Hubbard, T. J., Chothia, C., and Murzin, A. G., 2004. SCOP Database in 2004: Refinements Integrate Structure and Sequence Family Data. Nucleic Acids Research, 32(suppl_1):D226-D229.

Bairoch, A., and Apweiler, R., 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic acids research, 28(1): 45-48.

Baldi, P. 2012. Autoencoders, Unsupervised Learning, and Deep Architectures. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning, pp. 37-49.

Baloğlu, U., B., 2006. Extraction of frequent patterns and potential motifs from dna sequences with data mining method. Fırat Üniverstesi, Fen Bilimleri Enstitüsü.

Bandyopadhyay, S., and Pal, S. K., 2007. Classification and Learning Using Genetic Algorithms: Applications in Bioinformatics and Web Intelligence. Springer-Verlag Berlin Heidelberg, India, 311.

Baştürk, A., Yüksel, M. E., Çalışkan, A., and Badem, H. 2017. Deep Neural Network Classifier for Hand Movement Prediction. In 2017 25th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

Bayrak, Ş., Takçı, H., and Eminli, M., 2012. Makine Oğrenme Yöntemleriyle N-Gram Tabanlı Dil Tanıma N-Gram Based Language Identification with Machine Learning Methods. ELECO '2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu, 534- 538.

Bengio, Y. 2009. Learning Deep Architectures for AI. Found. trends® Mach. Learn., 2(1), pp. 1–127.

Benhammou, Y., Achchab, B., Herrera, F. and Tabik, S. 2020. Break His Based Breast Cancer Automatic Diagnosis Using Deep Learning: Taxonomy, Survey and Insights. Neurocomputing, 375(2020): 9-24.

Ben-Hur, A., and Brutlag, D. 2003. Remote Homology Detection: a Motif Based Approach. Bioinformatics, 19(suppl_1): i26-i33.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Sayers, E. W. 2009. GenBank. Nucleic acids research, 37(suppl_1), D26-D31.

Berman, D. S., Buczak, A. L., Chavis, J. S., and Corbett, C. L., 2019. A survey of deep learning methods for cyber security. Information, 10(4): 122.

Bhavsar, H., and Ganatra, A. 2012. A Comparative Study of Training Algorithms for Supervised Machine Learning. International Journal of Soft Computing and Engineering (IJSCE), 2(4): 2231-2307.

biopyhton. [Online]. Available:https://biopython.org/

Blei, D. M., Ng, A. Y., and Jordan, M. I., 2003. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3(Jan): 993-1022.

Boser, B. E., Guyon, I. M., and Vapnik, V. N., 1992. A Training Algorithm for Optimal Margin Classiiers. In Proceedings of the fifth annual workshop on Computational learning theory, 144–152.

Bouckaert, R. R. 2004, December. Naive Bayes Classifiers That Perform Well with Continuous variables. In Australasian Joint Conference on Artificial Intelligence, (pp. 1089-1094). Springer, Berlin, Heidelberg.

Bradley A.: The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition 30 (7) (1997): 1145-1159.

Braga-Neto, U. M., and Dougherty, E. R. 2004. Is Cross-validation Valid for Small-sample Microarray Classification?. Bioinformatics, 20(3): 374-380.

Caliskan, A., Badem, H., Basturk, A., and Yuksel, M. E. 2017. Diagnosis of the Parkinson Disease by Using Deep Neural Network Classifier. Istanbul University-Journal of Electrical & Electronics Engineering, 17(2):3311-3318.

Center Berkeley, 2016. Caffe. (Online).Available: http://caffe.berkeleyvision.org/

Cha, S. H. 2007. Comprehensive Survey on Distance/similarity Measures between Probability Density Functions. City, 1(2), 1.

Chandonia, J. M., Fox, N. K., and Brenner, S. E., 2018. SCOPe: Classification of Large Macromolecular Structures in the Structural Classification of Proteins—Extended Database. Nucleic acids research, .

Chang, Y. W., Hsieh, C. J., Chang, K. W., Ringgaard, M., and Lin, C. J. 2010. Training and Testing Low-degree Polynomial Data Mappings via Linear SVM. Journal of Machine Learning Research, 11(4).

Chay, Z., E., Lee, C., H., Lee, K., C., 2010. Oon J S. Ling M H. Russel and Rao Coefficient is a Suitable Substitute for Dice Coefficient in Studying Restriction Mapped Genetic Distances of Escherichia Coli. Computational and Mathematical Biology, 1(1): 1-9.

Chen, J., Guo, M., Li, S., and Liu, B., 2017. ProtDec-LTR2. 0: an Improved Method for Protein Remote Homology Detection by Combining Pseudo Protein and Supervised Learning to Rank. Bioinformatics, 33(21): 3473-3476.

Chen, J., Guo, M., Wang, X., and Liu, B., 2018. A Comprehensive Review and Comparison of Different Computational Methods for Protein Remote Homology Detection. Briefings in Bioinformatics, 19(2): 231-244.

Chen, J., Liu, B., and Huang, D., 2016. Protein Remote Homology Detection Based on an Ensemble Learning Approach. BioMed Research International, 2016: 5813645.

Cheng, B. Y. M., Carbonell, J. G., and Klein‐Seetharaman, J. 2005. Protein Classification Based on Text Document Classification Techniques. Proteins: Structure, Function, and Bioinformatics, 58(4), 955-970.

Chollet, F., 2018. Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y., 2015. Attention-based Models for Speech Recognition. In Advances in Neural Information Processing Systems, 2015:577-585.

Chowdhary, S., Sharma, V., Patki, A. B., and Gaur, V., 2010. Role And Scope of Support Vector Machine in Bioinformatics. Proceedings of the 4th National Conference, INDIACom-2010.

Christian Theil H., and Juhl Jensen, L., J., 2013.Are Graph Databases Ready for Bioinformatics?. Bioinformatics, 29(24): 3107.

Collobert, R., Kavukcuoglu, K., and Farabet, C. 2016. Torch. (Online). Available: http://http:// torch.ch/

Damoulas, T., and Girolami, M. A. 2008. Probabilistic Multi-class Multi-kernel Learning: on Protein Fold Recognition and Remote Homology Detection. Bioinformatics, 24(10): 1264-1270.

De Dieu Uwisengeyimana, J., and Ibrikci, T. 2017. Diagnosing Knee Osteoarthritis Using Artificial Neural Networks and Deep Learning. Biomedical Statistics and Informatics, 2(3): 95.

Dean, J., and Ghemawat, S., 2008. MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 51(1): 107-113.

Dempster, A., Laird, N. M., and Rubin, D., 1977. Maximum Likelihood from Incomplete Data via the EM algorithm. Journal of the Royal Statistical Society B, 39:1–38.

Deng, Z., Zhu, X., Cheng, D., Zong, M., and Zhang, S. 2016. Efficient kNN Classification Algorithm for Big Data. Neurocomputing, 195: 143-148.

Dong, Q. W., Wang, X. L., and Lin, L. 2006. Application of Latent Semantic Analysis to Protein Remote Homology Detection. Bioinformatics, 22(3):285-290.

dos Santos, C., and Gatti, M., 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014:69-78.

Duda, R.O., and Hart, P.E. 1973. Bayes Decision Theory. Chapter 2 in Pattern Classification and Scene Analysis, pp. 10–43. John Wiley.

El-Din, D. M. 2016. Enhancement Bag-of-words Model for Solving the Challenges of Sentiment Analysis. International Journal of Advanced Computer Science and Applications, 7(1).

Endres, F., Plagemann, C., Stachniss, C., ans Burgard, W., 2009. Unsupervised Discovery of Object Classes from Range Data Using Latent Dirichlet Allocation. Robotics: Science and Systems V, 2009:Paper15.

Eskin, E., Weston, J., Noble, W. S., and Leslie, C. S., 2003. Mismatch String Kernels for SVM Protein Classification. In Advances in Neural Information Processing Systems, 2003:1441-1448.

Faisal, M., and Zamzami, E. M. 2020, June. Comparative Analysis of Inter-Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance. In Journal of Physics: Conference Series, 1566(1):012112). IOP Publishing.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y., 2013. Learning Hierarchical Features for Scene Labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1915-1929.

Fawcett, T. 2006. An introduction to ROC analysis. Pattern Recognition letters, 27(8):861-874.

Fiannaca, A., La Rosa, M., La Paglia, L., Messina, A., and Urso, A., 2016. BioGraphDB: a new GraphDB Collecting Heterogeneous Data for Bioinformatics Analysis. Proceedings of BIOTECHNO.

Fix E, Hodges J L. Discriminatory Analysis. Nonparametric Discrimination; Consistency Properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, TX, USA. 1951.

Frezza, H., 2013. Support Vector Machines Tutorial. Frezza. Buet@ supe. lec. fr, (Accessed Mar., 2014).

Fukushima, K., and Miyake, S. 1982. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Visual Pattern Recognition. In Competition and Cooperation in Neural Nets (pp. 267-285). Springer, Berlin, Heidelberg.

Furat, F. G., and Ibrikci, T. 2017 Tumor Type Detection Using Naive Bayes Algorithm on Gene Expression Cancer Rna Seq Data Set. International Conference on Engineering Technologies (ICENTE'17), 482-486.

Ganapathiraju, M., Weisser, D., Rosenfeld, R., Carbonell, J. G., Reddy, R., and Klein-Seetharaman, J. 2002. Comparative ngram Analysis of whole-genome Sequences.

Ganganwar, V. 2012. An Overview of Classification Algorithms for Imbalanced Datasets. International Journal of Emerging Technology and Advanced Engineering, 2(4): 42-47.

Gemci Furat F., and İbrikçi T., 2017. Tumor Type Detection Usıng Naive Bayes Algorithm on Gene Expression Cancer Rna Seq Data Set. ICENTE' 17, 482-486.

Gemci, F., and Ibrikci, T. 2019. Using Deep Learning Algorithm to Diagnose Parkinson Disease wİth High Accuracy. Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi, 22, 19- 25.

Google, (2016). Tensorflow. (Online). Available: https://www.tensorflow.org/

Gouvier, W.D., Hayes, J.S. and Smiroldo, B.B. 1998. The Significance of Base Rates, Test Sensitivity, Test Specificity, and Subjects Knowledge of Symptoms in Assessing TBI Sequelae and Malingering. In C.R. Reynolds (Ed.), Detection of Malingering During Head Injury Litigation (pp. 55–80). New York: Plenum Press.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., and Chen, T., 2018. Recent Advances in Convolutional Neural Networks. Pattern Recognition, 77: 354-377.

Guimaraes, V., Hondo, F., Almeida, R., Vera, H., Holanda, M., Araujo, A., Walter, M., E., and Lifschitz, S. A Study of Genomic Data Provenance in NoSQL Document-oriented Database systems. In Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on. IEEE, 2015:1525-1531.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. 2016. Deep Learning for Visual Understanding: A Review. Neurocomputing, 187:27-48.

Hagan, M. T., Demuth, H. B., and Beale, M. H. 1996. Neural Network Design. Boston, MA: PWS Publishing. https://doi.org/10.1007/BF00738424

Han, H., Wang, W-Y., and Mao, B-H., 2005. Borderline-smote: a New Over-sampling Method in Imbalanced Data Sets Learning. In: Proceedings of International Conference on Intelligent Computing. Springer, Berlin, pp 878–887.

Han, J., Pei, J., and Kamber, M. 2011. Data Mining: Concepts and Techniques. Elsevier.

He, H., Bai, Y., Garcia, E. A., and Li, S. 2008, June. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In 2008 IEEE international joint Conference on Neural Networks (IEEE world congress on computational intelligence) (pp. 1322-1328). IEEE.

Hennekens, C.H., and Buring, J.E. 1987. Epidemiology in Medicine. Boston, MA: Little, Brown and Company.

Hermans, M., and Schrauwen, B., 2013. Training and Analysing Deep Recurrent Neural Networks. In Advances in Neural Information Processing Systems, 2013:190-198.

Hetal Bhavsar and Amit Ganatra, 2012, September. A Comparative Study of Training Algorithms for Supervised Machine Learning. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231- 2307, 2(4).

Hinton, G. E. and Salakhutdinov, R. R. 2006. Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786): 504–507.

Hinton, G. E. and Sejnowski, T. J. 1983. Optimal Perceptual Inference. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Hashington DC, pp. 448-453.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term Memory. Neural Computation, 9(8): 1735-1780.

Hofmann, T., 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning, 42(1-2): 177-196.

Holloway, E., 2020. Tutorial: Bioinformatics Basics. Communications of the Blyth Institute, 2(2): 35-38.

Holmberg, K., and Hallander, H. O., 1973. Numerical taxonomy and laboratory identification of Bacterionema matruchotii, Rothia dentocariosa, Actinomyces naeslundii, Actinomyces viscosus, and some related bacteria. Microbiology, 76(1): 43-63.

http://api.mongodb.com

http://www.buyukveri.co/category/mongo/

https://aminoacidsguide.com/

https://www.tutorialspoint.com/mongodb/mongodb_map_reduce.htm

Hubel, D. H. and Wiesel, T. N. 1962. Receptive Fields, Binocular Interaction and Functional Architecture in the cat's Visual Cortex. J. Physiol., vol. 160, no. 1, pp. 106–154.

Ivakhnenko, A. G. and Lapa, V. G. 1965. Cybernetic Predicting Devices. CCM Information Corporation.

J. Han and M. Kamber, Data Mining Concepts and Techniques, Elsevier, 2011.

Jarad, A., Katkar, R., Shaikh, A. R., and Salve, A., 2015. Intelligent Heart Disease Prediction System with MONGODB. Future, 13, 14.

Jiang, L., Cai, Z., Zhang, H., and Wang, D. 2013. Naive Bayes Text Classifiers: a Locally Weighted Learning Approach. Journal of Experimental & Theoretical Artificial Intelligence, 25(2): 273-286.

Jiang, L., Li, C., Wang, S., and Zhang, L. 2016. Deep Feature Weighting for Naive Bayes and its Application to Text Classification. Engineering Applications of Artificial Intelligence, 52: 26-39.

Karabulut, E. M., and Ibrikci, T. 2015. Emphysema Discrimination from raw HRCT Images by Convolutional Neural Networks. In 2015 9th International Conference on Electrical and Electronics Engineering (ELECO) (pp. 705-708). IEEE.

Karabulut, E. M., and Ibrikci, T. 2016, April. Texture Analysis of Melanoma Images for Computer-aided Diagnosis. In Int. Conference on Intelligent Computing, Computer Science & Information Systems (ICCSIS 16) 2:26-29.

Karabulut, E. M., and Ibrikci, T. 2017. Discriminative Deep Belief Networks for Microarray Based Cancer Classification.

Karakuş, B. 2018. Derin Öğrenme ve Büyük Veri Yaklaşımları ile Metin Analizi. Elazığ: Fırat Üniversitesi Doktora Tezi.

Karasoy, O., and Ballı, S., 2016. Google Maps ve Genetik Algoritmalarla GSP Çözümü İçin Öneri. XVIII. AKADEMİK BİLİŞİM KONFERANSI -- AB 2016, 2016.

Karplus, K., Barrett, C., and Hughey, R., 1998. Hidden Markov Models for Detecting Remote Protein Homologies. Bioinformatics (Oxford, England), 14(10):846-856.

Kayaalp, K., and Süzen, A. A. 2018 DERİN ÖĞRENME.

KIZRAK, M. A., and BOLAT, B. 2018. Derin öğrenme ile Kalabalık Analizi Üzerine Detaylı bir Araştırma. Bilişim Teknolojileri Dergisi, 11(3): 263-286.

Kim, S. B., Rim, H. C., Yook, D., and Lim, H. S. 2002, August. Effective Methods for Improving Naive Bayes Text Classifiers. In Pacific Rim International Conference on Artificial Intelligence (pp. 414-423). Springer, Berlin, Heidelberg.

Kingma, D. P., and Ba, J., 2014. Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1412.6980.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S., 2015. Skip-thought vectors. In Advances in Neural Information Processing Systems pp. 3294-3302.

Kocher, M., and Savoy, J., 2017. Distance Measures in Author Profiling. Information Processing & management, 53(5): 1103-1119.

Kohavi, R., 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. Int. Jt. Conf. Artif. Intell. Available at: http://robotics.stanford.edu/~ronnyk (Accessed January 2, 2021)

Komiya, K., Sato, N., Fujimoto, K., and Kotani, Y., 2011, September. Negation Naive Bayes for Categorization of Product Pages on the Web. In Proceedings of the International Conference Recent Advances in Natural Language Processing 2011 (pp. 586-591).

Kruengkrai, C., and Jaruskulchai, C., 2002. A Parallel Learning Algorithm for Text Classification. In Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 201-206). ACM.

Kumar, G. R., Mangathayaru, N., and Narasimha, G. 2015, September. Intrusion Detection Using Text Processing Techniques: a Recent Survey. In Proceedings of the The International Conference on Engineering & MIS 2015 (pp. 1-6).

Kumar, N. P., Rao, M. V., Krishna, P. R., and Bapi, R. S. 2005, December. Using Sub-sequence Information with kNN for Classification of Sequential Data. In International Conference on Distributed Computing and Internet Technology (pp. 536-546). Springer, Berlin, Heidelberg.

L. Jiang et al. 2016. Deep Feature Weighting for Naive Bayes and its Application to Text Classification. Engineering Applications of Artificial Intelligence 52:26–39.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L., 1989. Handwritten Digit Recognition with a Back-propagation Network. Advances in Neural Information Processing Systems, 2.

Leinonen, R., Nardone, F., Zhu, W., and Apweiler, R., 2006. UniSave: the UniProtKB sequence/annotation version database. Bioinformatics, 22(10): 1284-1285.

Leslie, C., Eskin, E., and Noble, W. S., 2001. The Spectrum Kernel: A String Kernel for SVM Protein Classification. In Biocomputing, 2002:564-575.

Li, J., Luong, M. T., and Jurafsky, D., 2015. A Hierarchical Neural Autoencoder for Paragraphs and Documents. 2015:1106–1115.

Li, Y., and Manoharan, S., 2013. A Performance Comparison of SQL and NoSQL Databases. In 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) (pp. 15-19). IEEE.

Liao, L., and Noble, WS., 2003. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. Journal of computational biology, 10(6): 857-868.

Lingner, T., and Meinicke, P., 2008. Word Correlation Matrices for Protein Sequence Analysis and Remote Homology Detection. BMC Bioinformatics, 9(1): 259.

Liu, B., Chen, J., and Wang, X., 2015. Application of Learning to Rank to Protein Remote Homology Detection. Bioinformatics, 31(21): 3492-3498.

Liu, B., Lee, W. S., Yu, P. S., and Li, X., 2002. Partially Supervised Classification of Text Documents. In ICML, Vol. 2: 387-394.

Liu, B., Liu, B., Liu, F., and Wang, X., 2014. Protein Binding Site Prediction by Combining Hidden Markov Support Vector Machine and Profile-based Propensities. The Scientific World Journal, 2014.

Liu, B., Wang, X., Lin, L., Dong, Q., and Wang, X., 2008. A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis. BMC bioinformatics, *9*(1): 1-16.

Liu, B., Yi, J., Aishwarya, S. V., Lan, X., Ma, Y., Huang, T. H., ... and Jin, V. X., 2013. QChIPat: a Quantitative Method to Identify Distinct Binding Patterns for two Biological ChIP-seq Samples in Different Experimental Conditions. Bmc Genomics, 14(8): S3.

Liu, B., Zhang, D., Xu, R., Xu, J., Wang, X., Chen, Q., ... and Chou, K. C., 2014. Combining Evolutionary Information Extracted from Frequency Profiles with Sequence-based Kernels for Protein Remote Homology Detection. Bioinformatics, 30(4): 472-479.

Longadge, R., and Dongre, S. 2013. Class Imbalance Problem in Data Mining Review. arXiv preprint arXiv:1305.1707.

Lovato, P., Cristani, M., and Bicego, M., 2016. Soft Ngram representation and modeling for protein remote homology detection. IEEE/ACM transactions on computational biology and bioinformatics, 14(6): 1482-1488.

Lovato, P., Giorgetti, A., and Bicego, M., 2015. A Multimodal Approach for Protein Remote Homology Detection. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 12(5): 1193-1198.

Luscombe, N. M., Greenbaum, D., and Gerstein, M., 2001. What is Bioinformatics? A Proposed Definition and Overview of the Field. Methods of Information in Medicine, 40(04): 346-358.

Mallat, S., 2016. Understanding Deep Convolutional Networks. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065).

Matic, N., Guyon, I., Denker, J., and Vapnik, V. 1993. Writer-adaptation for On-line Handwritten Character Recognition. In Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93) pp. 187-191. IEEE.

McCallum, A., and Nigam, K., 1998, July. A Comparison of Event Models for Naive Bayes Text Classification. In AAAI-98 Workshop on Learning for Text Categorization (Vol. 752, No. 1, pp. 41-48).

McCulloch, W. S., and Pitts, W., 1943. A Logical Calculus of the Idea Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 5: 115–133.

Méndez, J. R., Cid, I., Glez-Peña, D., Rocha, M., and Fdez-Riverola, F., 2008, July. A Comparative Impact Study of Attribute Selection Techniques on Naive Bayes Spam filters. In Industrial Conference on Data Mining (pp. 213-227). Springer, Berlin, Heidelberg.

Metin, İ. A., and Karasulu, B. 2015. İnsan Aktivitelerinin Sınıflandırılmasında Tekrarlayan Sinir Ağı Kullanan Derin Öğrenme Tabanlı Yaklaşım. Veri Bilimi, 2(2): 1-10.

Min, S., Lee, B., and Yoon, S., 2017. Deep Learning in Bioinformatics. Briefings in Bioinformatics, 18(5): 851-869.

Mishra O., Lodhi P., and Mehta S., 2018. Document Oriented NoSQL Databases: An Empirical Study In International Conference on Recent Developments in Science, Engineering and Technology, Springer, Singapore, 2017:126-136.

MongoDB. [Online]. Available: http://www.mongodb.org/.

Mousa, A., and Yusof, Y., 2018. Fuzzy C-Means with Improved Chebyshev Distance for Multi-Labelled Data. Journal of Engineering and Applied Sciences,13(2): 353-360.

Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C., 1995. SCOP: a Structural Classification of Proteins Database for the Investigation of Sequences and Structures. Journal of Molecular Biology, 247(4): 536-540.

Nakshathram, S., Duraisamy, R., and Pandurangan, M., 2021. Sequence-Order Frequency Matrix-Sampling and Machine learning with Smith-Waterman (SOFM-SMSW) for Protein Remote Homology Detection.

Needleman, S. B., and Wunsch, C. D., 1970. A General Method Applicable to the Search for Similarities in the Amino acid Sequence of Two Proteins. Journal of Molecular Biology, 48(3): 443-453.

Nielsen, M. A., 2015. Neural Networks and Deep Learning. Vol. 25. http://neuralnetworksanddeeplearning.com/, USA, 224s.

Ocaña, M. I. G., Román, K. L. L., Urzelai, N. L., Ballester, M. Á. G., and Oliver, I. M., 2020. Medical Image Detection Using Deep Learning. In Deep Learning in Healthcare (pp. 3-16). Springer, Cham.

Oliver, S., and Majchrzak, T., A., 2012. Using Document-based Databases for Medical Information Systems in Unreliable Environments. 9th International ISCRAM Conference. 2012:23.

O'Shea, K., and Nash, R. 2015. An Introduction to Convolutional Neural Networks. arXiv preprint arXiv:1511.08458.

ÖZTÜRK, S., and ATMACA, H., E., 2017. İlişkisel ve İlişkisel Olmayan (NoSQL) Veri tabanı Sistemleri Mimari Performansının Yönetim Bilişim Sistemleri Kapsamında İncelenmesi. Bilişim Teknolojileri Dergisi, 10(2): 199-209.

Palaniappan, S., and Awang, R. 2008, March. Intelligent Heart Disease Prediction System Using Data Mining Techniques. In 2008 IEEE/ACS International Conference on Computer Systems and Applications (pp. 108-115). IEEE.

Papageorgiou, C. P., Oren, M., and Poggio, T., 1998. A General Framework for Object Detection. Sixth International Conference on Computer Vision (IEEE), 1998:555-562.

Pattekari, S. A., and Parveen, A. 2012. Prediction System for Heart Disease Using Naïve Bayes. International Journal of Advanced Computer and Mathematical Sciences, 3(3): 290-294.

Pearson, W. R., 1991. Searching Protein Sequence Libraries: Comparison of the Sensitivity and Selectivity of the Smith-Waterman and FASTA Algorithms. Genomics, 11(3): 635-650.

PostgreSQL. [Online]. Available: https://www.postgresql.org/

Prasath, V B., Alfeilat H A A, Hassanat, A., Lasassmeh, O., Tarawneh, A S., Alhasanat, M B., and Salman, H S E., 2017. Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier--A Review. arXiv preprint arXiv:1708.04321.

Qaiser, S., and Ali, R., 2018. Text Mining: use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications, 181(1): 25-29.

Radivojac, P., 2022. Advancing remote homology detection: A step toward understanding and accurately predicting protein function. Cell Systems, 13(6): 435-437.

Raschka, S., 2014. Naive Bayes and Text Classification I-introduction and Theory. arXiv preprint arXiv:1410.5329.

Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., and Yang, G. Z., 2016. Deep Learning for Health Informatics. IEEE Journal of Biomedical and Health Informatics, 21(1):4-21.

Religia, Y., and Sunge, A. S. 2019, March. Comparison of Distance Methods in K-Means Algorithm for Determining Village Status in Bekasi District. In 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT) (pp. 270-276). IEEE.

Rennie, J. D., Shih, L., Teevan, J., and Karger, D. R., 2003. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In Proceedings of the 20th International Conference on Machine Learning (ICML-03) (pp. 616-623).

Rost. B., 1999. Twilight Zone of Protein Sequence Alignments, Protein Eng, vol. 12 (pg. 85-94)

Ruder, S., 2016. An Overview of Gradient Descent Optimization Algorithms. arXiv preprint arXiv:1609.04747.

S.-J. Yen, and Y.-S. Lee, 2009. Cluster-based Under-sampling Approaches for Imbalanced Data Distributions Expert Systems with Applications, 36: 5718–5727.

Sak, H., Senior, A. W., and Beaufays, F., 2014. Long Short-term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling.

Salakhutdinov, R. and Hinton, G., 2009. Semantic Hashing. Int. J. Approx. Reason., 50: 969–978.

Sarıkaya, M. A. and İnce, G. 2017. Emotion Recognition from EEG Signals Through One Electrode Device. In Signal Processing and Communications Applications Conference (SIU), 25th (pp. 1-4). IEEE.

Saud, S., Jamil, B., Upadhyay, Y., and Irshad, K., 2020. Performance Improvement of Empirical Models for Estimation of Global Solar Radiation in India: a k-

fold Cross-validation Approach. Sustainable Energ. Tech. Assessments 40, 100768. doi:10.1016/j.seta.2020.100768.

Schuster, M., and Paliwal, K. K., 1997. Bidirectional Recurrent Neural Networks. IEEE Transactions on Signal Processing, 45(11): 2673-2681.

Schwarz, M., Lobur, M., and Stekh, Y., 2017, February. Analysis of the Effectiveness of Similarity Measures for Recommender Systems. In 2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM) (pp. 275-277). IEEE.

SCOP. [Online]. Available:http://scop.berkeley.edu/

Shah, A. R., Oehmen, C. S., and Webb-Robertson, B. J., 2008. SVM-HUSTLE—an Iterative Semi-supervised Machine Learning Approach for Pairwise Protein Remote Homology Detection. Bioinformatics, 24(6):783-790.

Shao, B., and Conrad, T. O., 2015. Are NoSQL Data Stores Useful for Bioinformatics Researchers?. International Journal on Recent and Innovation Trends in Computing and Communication, 3(3): 1704-1708.

Shao, J., Chen, J., and Liu, B., 2021. ProtRe-CN: Protein Remote Homology Detection by Combining Classification Methods and Network Methods via Learning to Rank. IEEE/ACM Transactions on Computational Biology and Bioinformatics.

Singh Chauhan, A., Kedawat, A., and Pooja Parnami, P., 2015. An Approach to Implement Map Reduce with NoSQL Databases. International Journal Of Engineering And Computer Science, 4(8):13635-13639.

Smiti, S., and Soui, M., 2020. Bankruptcy Prediction Using Deep Learning Approach Based on Borderline SMOTE. Information Systems Frontiers, 22(5), 1067-1083.

Stabili D, Marchetti M, and Colajanni M. 2017, September. Detecting Attacks to Internal Vehicle Networks Through Hamming Distance. In AEIT International Annual Conference (pp. 1-6). IEEE.

Şeker, A., Diri, B., and Balık, H. H., 2017. Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında bir İnceleme. Gazi Mühendislik Bilimleri Dergisi (GMBD), 3(3): 47-64.

Tensorflow. [Online]. Available: https://www.tensorflow.org

Thai-Nghe N, Do TN, and Schmidt-Thieme L, 2000. Learning Optimal Threshold on Resampling Data to Deal with Class Imbalance. Proc. of the 8th IEEE International Conference on Computing.

Theano. (Online). Available: http://deeplearning.net/software/ theano/

Theodoridis, S., and Koutroumbas, K., 1999, July. Pattern Recognition and Neural Networks. In Advanced Course on Artificial Intelligence (pp. 169-195). Springer, Berlin, Heidelberg.

Tim D., 2015. Deep Learning in a Nutshell: History and Training Parallel Forall. (Online). Available: https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/. (Accessed: 20-Mar-2017).

Tomović, A., Janičić, P., and Kešelj, V., 2006. n-Gram-based Classification and Unsupervised Hierarchical Clustering of Genome Sequences. Computer Methods and Programs in Biomedicine, 81(2): 137-153.

Tripathy, A., and Rath, S. K. 2017. Classification of Sentiment of Reviews Using Supervised Machine Learning Techniques. International Journal of Rough Sets and Data Analysis (IJRSDA), 4(1): 56-74.

Wu, G., Bazer, F. W., Dai, Z., Li, D., Wang, J., and Wu, Z. 2014. Amino acid nutrition in animals: protein synthesis and beyond. Annual Review of Animal Biosciences, 2(1): 387-417.

Yan, H., Zhou, X., and Ge, Y., 2015, December. Neighborhood Repulsed Correlation Metric Learning for Kinship Verification. In 2015 Visual Communications and Image Processing (VCIP) (pp. 1-4). IEEE.

Yeh, J. H., and Chen, C., H., 2010. Protein Remote Homology Detection Based on Latent Topic Vector Model. In 2010 International Conference on Networking and Information Technology, (pp. 456-460) IEEE.

You, Z. H., Yu, J. Z., Zhu, L., Li, S., and Wen, Z. K., 2014. A MapReduce Based Parallel SVM for Large-scale Predicting Protein–protein Interactions. Neurocomputing, 145: 37-43.

Yousef, M., Jung, S., Kossenkov, A. V., Showe, L., C., and Showe, M., K., (2007). Naïve Bayes for MicroRNA Target Predictions—Machine Learning for MicroRNA Targets. Bioinformatics, 23(22): 2987-2992.

Zeng, X., and Martinez, T. R., 2000. Distribution-balanced stratified cross-validation for accuracy estimation. Journal of Experimental & Theoretical Artificial Intelligence, 12(1): 1-12.

Zhang, W., Yoshida, T., and Tang, X., 2011. A comparative study of TF* IDF, LSI and Multi-words for text classification. Expert Systems with Applications, 38(3): 2758-2765

Zhang, Y., Jin, R., and Zhou, Z. H., 2010. Understanding Bag-of-words Model: a Statistical Framework. International Journal of Machine Learning and Cybernetics, 1(1): 43-52.

Zhao, R., and Mao, K., 2017. Fuzzy Bag-of-words Model for Document Representation. IEEE Transactions on Fuzzy Systems, 26(2): 794-804.

Zhihao, P., Fenglong, Y., and Xucheng, L., 2019, April. Comparison of the Different Sampling Techniques for Imbalanced Classification Problems in Machine Learning. In 2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (pp. 431-434). IEEE.

Zou, Q., Hu, Q., Guo, M., and Wang, G., 2015. HAlign: Fast Multiple Similar DNA/RNA Sequence Alignment Based on the Centre Star Strategy. Bioinformatics, 31(15): 2475-2481.

CURRICULUM VITAE

Fahriye Gemci has graduated from Selcuk University in department of Computer Engineering in 2010 with B.S degree. She received the MS. degree in Electronics and Computer Engineering in 2015. She is currently working as a research assistant in Computer Enginnering Department at Kahramanmaras Sutcu Imam University. Her research interests are artificial intelligence, data mining, bioinformatics and social media.

APPENDIX

APPENDIX A

Table A.1.The experimental results for detection of remote homologous proteins using deep learning

Family ID	AUROC Score	Accuracy	mae	mse	F1 score	Presicion Score	Recall score	Kappa score
1.4.1.1	0.9725	0.9742	0.0258	0.0258	0.9807	0.2711	0.9130	0.4371
1.4.1.2	0.8916	0.9116	0.0885	0.0885	0.9445	0.0915	0.875	0.1672
1.4.1.3	0.9487	0.9392	0.0608	0.0608	0.9603	0.1579	1.0	0.2581
1.27.1.1	0.9336	0.9807	0.0193	0.0193	0.9872	0.1309	0.8333	0.2580
1.27.1.2	0.8851	0.8449	0.1551	0.1551	0.9120	0.0260	1.0	0.0429
1.36.1.2	0.8728	0.8842	0.1158	0.1158	0.9311	0.0377	0.7143	0.0783
1.36.1.5	0.8707	0.8635	0.1365	0.1365	0.9193	0.0463	0.8846	0.0825
1.41.1.2	0.9390	0.9871	0.0129	0.0129	0.9892	0.3488	0.8333	0.5498
1.41.1.5	0.9241	0.9536	0.0463	0.0463	0.9690	0.1109	0.76	0.2280
1.45.1.2	0.9434	0.9940	0.0060	0.0060	0.9944	0.5223	0.8333	0.7113
2.1.1.1	0.8761	0.8899	0.1101	0.1101	0.9215	0.1247	0.7097	0.2315
2.1.1.2	0.8850	0.9821	0.0180	0.0180	0.9828	0.5117	0.7727	0.6991
2.1.1.3	0.8109	0.7845	0.2156	0.2156	0.8561	0.0763	0.75	0.1200
2.1.1.4	0.8773	0.9316	0.0684	0.0684	0.9423	0.0583	0.2727	0.1519
2.1.1.5	0.8709	0.9018	0.0982	0.0982	0.9276	0.0985	0.5556	0.2069
2.5.1.1	0.8778	0.9323	0.0677	0.0677	0.9600	0.0266	0.5455	0.0722
2.5.1.3	0.8973	0.9460	0.0541	0.0541	0.9674	0.0382	0.6	0.1001
2.9.1.2	0.8982	0.9384	0.0616	0.0616	0.9622	0.0583	0.7143	0.1307
2.9.1.3	0.9658	0.9572	0.4280	0.4280	0.9728	0.1429	1.0	0.2405
2.9.1.4	0.9454	0.9380	0.0620	0.0620	0.9622	0.1031	1.0	0.1763
2.28.1.1	0.8796	0.9686	0.0314	0.0314	0.9722	0.0373	0.2046	0.1414
2.28.1.3	0.8261	0.7553	0.2447	0.2447	0.8476	0.0551	1.0	0.0795
2.38.4.1	0.8600	0.9142	0.0858	0.0858	0.9479	0.0202	0.4	0.0562
2.38.4.3	0.8378	0.9154	0.0846	0.0846	0.9485	0.0181	0.3636	0.0511
2.38.4.5	0.8408	0.8859	0.1141	0.1141	0.9323	0.0430	0.7777	0.0854
2.44.1.2	0.8472	0.9048	0.0952	0.0952	0.9182	0.0343	0.05	0.0103
2.52.1.2	0.8472	0.9086	0.0914	0.0914	0.9484	0.0092	0.4	0.0258
2.56.1.2	0.9501	0.9269	0.0731	0.0731	0.9581	0.0563	1.0	0.0992
3.1.8.1	0.9681	0.9528	0.0472	0.0472	0.9709	0.1177	1.0	0.2015
3.1.8.3	0.9319	0.9509	0.0491	0.0491	0.9697	0.0775	0.8	0.1608
3.2.1.2	0.7942	0.6527	0.3473	0.3473	0.7779	0.0339	1.0	0.0430
3.2.1.3	0.8119	0.8349	0.1651	0.1651	0.8992	0.0361	0.6666	0.0684
3.2.1.4	0.8645	0.8362	0.1638	0.1638	0.9001	0.0693	1.0	0.1093
3.2.1.5	0.9562	0.9321	0.0679	0.0679	0.9558	0.1522	1.0	0.2482
3.2.1.6	0. 8840	0.9122	0.0878	0.0878	0.9438	0.0535	0.6	0.1240
3.2.1.7	0.8953	0.8366	0.1634	0.1634	0.9003	0.0695	1.0	0.1096
3.3.1.2	0.8692	0.8133	0.1867	0.1867	0.8908	0.0345	1.0	0.0545
3.3.1.5	0.8897	0.9484	0.0517	0.0517	0.9678	0.0431	0.5625	0.1163
3.32.1.1	0.7926	0.8997	0.1003	0.1003	0.9374	0.0690	0.7778	0.1356
3.32.1.8	0.8808	0.9584	0.0416	0.0416	0.9688	0.0326	0.2727	0.1177
3.32.1.11	0.9216	0.8967	0.1033	0.1033	0.9360	0.1020	1.0	0.1675
3.32.1.13	0.8483	0.8607	0.1393	0.1393	0.9149	0.0484	0.75	0.0925
3.42.1.1	0.8591	0.9283	0.0718	0.0718	0.9548	0.0259	0.4	0.0762
3.42.1.5	0.7436	0.9028	0.0972	0.0972	0.9407	0.0153	0.3077	0.0380
3.42.1.8	0.7717	0.7864	0.2137	0.2137	0.8721	0.0280	0.8	0.0466

7.3.5.2	0.9030	0.9681	0.0319	0.0319	0.9797	0.0904	0.7777	0.1928
7.3.6.1	0.9744	0.9683	0.0318	0.0318	0.9777	0.2432	1.0	0.3812
7.3.6.2	0.9660	0.9745	0.0255	0.0255	0.9813	0.2705	0.9615	0.4257
7.3.6.4	0.9967	0.9857	0.0143	0.0143	0.9886	0.4166	1.0	0.5822
7.3.10.1	0.9436	0.9498	0.0502	0.0502	0.9614	0.3180	0.9579	0.4720
7.39.1.2	0.9750	0.9486	0.0514	0.0514	0.9686	0.1077	1.0	0.1853
7.39.1.3	0.7348	0.7974	0.2026	0.2026	0.8816	0.0126	0.5714	0.0221
7.41.5.1	0.8300	0.8879	0.1121	0.1121	0.9365	0.0103	0.4444	0.0257
7.41.5.2	0.8787	0.8133	0.1867	0.1867	0.8928	0.0233	1.0	0.0371
Average	**0.8863**	**0.9058**	**0.1014**	**0.1014**	**0.9405**	**0.1066**	**0.7484**	**0.1797**

APPENDIX B

Table B.0.1. Confusion matrixes of KNN with distances without cross validation

Distance/Similarity Methods	1.4.1.1 Family	2.1.1.2 Family	2.28.1.1 family	3.42.1.5 family	7.3.10.1 Family
Bray Curtis	[1994 0] [22 1]	[753 5] [14 8]	[3039 5] [44 0]	[1436 1] [13 0]	[3625 28] [44 51]
Euclidean	[1990 4] [10 13]	[750 8] [14 8]	[3040 4] [44 0]	[1435 2] [13 0]	[3618 35] [90 5]
Minkowski	[1990 4] [10 13]	[753 5] [5 17]	[3040 4] [44 0]	[1435 2] [13 0]	[3618 35] [90 5]
Dice	[1993 1] [23 0]	[757 1] [22 0]	[3041 3] [44 0]	[1437 0] [13 0]	[3641 12] [86 9]
Jaccard	[1993 1] [23 0]	[757 1] [22 0]	[3041 3] [44 0]	[1437 0] [13 0]	[3641 12] [86 9]
Chebyshev	[1990 4] [11 12]	[751 7] [18 4]	[3037 7] [44 0]	[1433 4] [13 0]	[3626 27] [87 8]
Cosine	[1994 0] [23 0]	[757 1] [22 0]	[3034 10] [44 0]	[1437 0] [13 0]	[3644 9] [67 28]
SokalSneath	[1993 1] [23 0]	[757 1] [22 0]	[3041 3] [44 0]	[1437 0] [13 0]	[3641 12] [86 9]
Correlation	[1994 0] [23 0]	[756 2] [22 0]	[3034 10] [44 0]	[1437 0] [13 0]	[3642 11] [67 28]
Matching	[1993 1] [23 0]	[757 1] [22 0]	[3043 1] [44 0]	[1436 1] [13 0]	[3590 63] [89 6]
Rogers Tanimoto	[1993 1] [23 0]	[758 0] [22 0]	[3043 1] [44 0]	[1436 1] [13 0]	[[3590 63] [89 6]
Sokal Michener	[1993 1] [23 0]	[757 1] [22 0]	[3043 1] [44 0]	[1436 1] [13 0]	[3590 63] [89 6]
Canberra	[1992 2] [23 0]	[757 1] [22 0]	[3043 1] [44 0]	[1436 1] [13 0]	[3589 64] [89 6]
Hamming	[1991 3] [23 0]	[758 0] [22 0]	[3044 0] [44 0]	[1437 0] [13 0]	[3552 101] [92 3]
Kulczynski	[1994 0] [23 0]	[758 0] [22 0]	[3043 1] [44 0]	[1437 0] [13 0]	[3653 0] [95 0]
RussellRao	[1994 0] [23 0]	[758 0] [22 0]	[3044 0] [44 0]	[1437 0] [13 0]	[3653 0] [95 0]

Table B.2. Confusion matrixes of KNN with distances for remote homology with cross validation with special k value fold

Distance/Similarity Methods	1.4.1.1 Family	2.1.1.2 Family	2.28.1.1 family	3.42.1.5 family	7.3.10.1 Family
Bray Curtis	[1981 13] [0 23]	[753 5] [1 21]	[3033 11] [2 42]	[1429 8] [2 11]	[3623 30] [5 90]
Euclidean	[1979 15] [1 22]	[753 5] [5 17]	[3028 16] [4 40]	[1425 12] [1 12]	[3610 43] [8 87]
Minkowski	[1979 15] [1 22]	[753 5] [5 17]	[3028 16] [4 40]	[1425 12] [1 12]	[3610 43] [8 87]
Dice	[1991 3] [3 20]	[755 3] [14 8]	[3043 1] [5 39]	[1437 0] [6 7]	[3631 22] [6 89]
Jaccard	[1991 3] [3 20]	[755 3] [14 8]	[3043 1] [5 39]	[1437 0] [6 7]	[3631 22] [6 89]
Chebyshev	[1985 9] [5 18]	[750 8] [13 9]	[3029 15] [5 39]	[1426 11] [6 7]	[3590 63] [11 84]
Cosine	[1993 1] [4 19]	[757 1] [17 5]	[3042 2] [4 40]	[1437 0] [13 0]	[3645 8] [17 78]
SokalSneath	[1991 3] [3 20]	[755 3] [14 8]	[3043 1] [5 39]	[1437 0] [6 7]	[3631 22] [6 89]
Correlation	[1993 1] [3 20]	[757 1] [17 5]	[3042 2] [4 40]	[1437 0] [13 0]	[3643 10] [17 78]
Matching	[1993 1] [6 17]	[758 0] [22 0]	[3024 20] [3 41]	[1436 1] [13 0]	[3634 19] [19 76]
Rogers Tanimoto	[1993 1] [6 17]	[758 0] [22 0]	[3024 20] [3 41]	[1436 1] [11 2]	[[3634 19] [19 76]
Sokal Michener	[1993 1] [6 17]	[758 0] [22 0]	[3024 20] [3 41]	[1436 1] [11 2]	[3634 19] [19 76]
Canbera	[1989 5] [6 17]	[758 0] [22 0]	[3025 19] [4 40]	[1435 2] [10 3]	[3637 16] [20 75]
Hamming	[1994 0] [10 13]	[758 0] [22 0]	[3030 14] [5 39]	[1436 1] [13 0]	[3639 14] [20 75]
Kulczynski	[1994 0] [14 9]	[758 0] [22 0]	[3044 0] [5 39]	[1437 0] [13 0]	[3653 0] [21 74]
Russell Rao	[1994 0] [18 5]	[758 0] [22 0]	[3044 0] [7 37]	[1437 0] [13 0]	[3653 0] [22 73]