T.C.

ALTINBAŞ UNIVERSITY

Electrical and Computer Engineering

ASSESSING THE QUALITY OF SCIENTIFIC ARTICLES USING ARTIFICIAL NEURAL NETWORKS

Iman Saad Khaleel

Master Thesis

Supervisor

Asst. Prof. Dr. Sefer KURNAZ

Istanbul, 2019

ASSESSING THE QUALITY OF SCIENTIFIC ARTICLES USING ARTIFICIAL NEURAL NETWORKS

by

Iman Saad Khaleel

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2019

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Sefer KURNAZ

Supervisor

Examining Committee Members

Academic Title Name SURNAME Faculty, University _____

Academic Title Name SURNAME Faculty, University _____

Academic Title Name SURNAME Faculty, University _____

I certify that this thesis satisfies all the requirements as a thesis for the degree of .....

Academic Title Name SURNAME

Head of Department

Academic Title Name SURNAME

Director

Approval Date of Graduate School of

Science and Engineering: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Iman Saad Khaleel

DEDICATION

ACKNOWLEDGMENTS

ABSTRACT

ASSESSING THE QUALITY OF SCIENTIFIC ARTICLES USING ARTIFICIAL NEURAL NETWORKS

Iman Saad Khaleel,

M.Sc., Electrical and Computer Engineering, Altınbaş University

Supervisor: Asst. Prof. Dr. Sefer KURNAZ

Date: Feb/2019

Pages: 57

With the recent digital revolution, many of the traditional services are being provided online, digitally. With the ease of access and the wide availability of the internet, more users of these services are gaining access, which imposes the need for handling such growth. One of the services that have been significantly affected by the digital revolution is scientific research. With the existence of digital libraries, researches from all around the world have been able to access the most recent articles, as soon as they are being published digitally. This ease of access has also encouraged many researchers to publish their works through these libraries, which has significantly increased the number of articles being submitted for publish.

Journals follow a certain procedure that requires experts to review these articles and ensure they match the requirement of the study field and the journal, before accepting these articles for publish. With the enormous number of submitted articles, this process is being extremely time-consuming. To improve the efficiency of this procedure, an estimation of the quality of the article can assist both the authors and reviewers making faster decisions. Authors can use this estimation to improve the quality of their articles before submitting them to the publisher, while the reviewers can produce a faster decision with the existence of such estimation. The quality of the paper can be estimated by recognizing the importance of the field the article is involved in and the quality of the writing. As these articles are

written in natural languages, measuring their quality is a complex process that requires complex machine learning techniques.

In this study, a novel technique is proposed to use artificial neural networks, according to their ability in Natural Language Processing (NLP), to estimate the quality of the article. The evaluation is conducted using three parts of the article, the title, keywords and abstract. In addition to the field of the study, the abstract provides an overview of the writing quality of the articles. Two approaches are used to measure the overall quality of the article. The first approach uses three neural networks, one per each component, to predict the overall quality. These measures are then fused into a single overall measure, using the average and median functions. The other approach uses a single hybrid neural network that processes all the three inputs simultaneously. However, the topology of the hybrid neural network consists of three separate networks, one per each component, that do not interconnect until a certain dense layer, so that, each network extracts information from a single component.

Different types of neural networks are evaluated when processing the title and abstract components, as the position of the words have a meaningful order, while the keywords are processed using a fully-connected dense network, as the order of keywords has no meaningful representation. The evaluation results show that the Convolutional Neural Network (CNN) has achieved the best performance among the other types of networks, Recurrent and LSTM networks. This performance is illustrated by means of lower Mean Squared Error (MSE) and average prediction time. Moreover, the proposed hybrid neural network has achieved the lowest MSE of 4.52 using the convolutional neural network. This result shows that this method can have a significant role in accelerating the submission process of scientific articles.

Keywords: Artificial Neural Networks; Convolutional Neural Networks; Recurrent Neural Network; Lon- Short-Term Memory; Natural Language Processing.

ÖZET

Yapay Sinir Ağları Kullanarak Bilimsel Makale Kalitesinin Değerlendirilmesi

Iman Saad Khaleel,

Yüksek Lisans, Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi

Danışman: Yrd. Prof. Dr. Sefer KURNAZ

Tarih: Şubat / 2019

Sayfalar: 57

Son zamanda gerçekleşen dijital devrim ile birlikte, geleneksel hizmetlerin çoğu internet üzerinden dijital olarak sağlanmaktadır. İnternetin yaygın bir şekilde kullanılabilir olması ve internete erişiminin kolaylığı ile daha fazla kullanıcının bu hizmetlere erişimini arttırıyor, Bu da böylesine bir büyümenin idare edilmesinin gerekliliğini gösterir. Dijital devrimden önemli ölçüde etkilenen hizmetlerden biri bilimsel araştırmadır. Dijital kütüphanelerin varlığıyla, dünyanın dört bir yanından araştırmacılar, dijital olarak yayınlanır yayınlanmaz en yeni makalelere erişebildi. Bu erişim kolaylığı aynı zamanda birçok araştırmacının çalışmalarını bu kütüphaneler aracılığıyla yayınlamasını teşvik etti ve bu da yayınlanmak üzere gönderilen makale sayısını önemli ölçüde arttırdı.

Dergiler, makalelerin yayınlanmasının kabul edilmesinden önce, çalışma alanı ile derginin gerekliliklerine eşleşmiş olmasını garantileyen ve uzmanların bu makaleleri gözden geçirmelerini gerektiren belirli bir prosedürü takip eder. Çok sayıda makalenin gönderilmesi ile bu işlemin çok zaman almasına neden olur. Bu prosedürün etkinliğini arttırmak için, makalenin kalitesine ilişkin bir değerlendirme hem yazarlara hem de hakemlere daha hızlı karar vermede yardımcı olabilir. Yazarlar makalelerini yayımcıya göndermeden önce makalelerinin kalitesini yükseltmek için bu değerlendirmeyi kullanabilirken, hakemler bu tahmininin varlığında daha hızlı kararlar verebilir. Makalenin kalitesi, makalenin içinde bulunduğu alanın önemi ve yazının kalitesi dikkate alınarak tahmin edilebilir. Bu makaleler doğal dillerde yazıldığından, kalitelerini ölçmek karmaşık makine öğrenme teknikleri gerektiren karmaşık bir süreçtir.

Bu çalışmada, makalenin kalitesini değerlendirmek için Doğal Dil İşleme'deki (NLP) yeteneklerine göre yapay sinir ağlarını kullanmak için yeni bir teknik önerilmiştir. Değerlendirme, makalenin üç kısmı; başlık, anahtar kelimeler ve özet, kullanılarak yapılır. Çalışma alanına ek olarak, özet, makalelerin yazma kalitesine genel bir bakış sunmaktadır. Makalenin genel kalitesini ölçmek için iki yaklaşım kullanılmaktadır. İlk yaklaşım, genel kaliteyi tahmin etmek için her bir bileşen için bir tane olmak üzere üç sinir ağı kullanır. Bu ölçümler daha sonra ortalama ve medyan fonksiyonlar kullanılarak tek bir genel ölçüye birleştirilir. Diğer yaklaşım, üç girişi de aynı anda işleyen tek bir hibrit sinir ağı kullanır. Bununla birlikte, hibrit sinir ağının topolojisi, her bir bileşen için bir tane olmak üzere, belirli bir yoğun katmana kadar birbirine bağlanamayan üç ayrı ağdan oluşur, böylece her bir ağ tek bir bileşenden bilgi alır.

Anahtar sözcüklerin sırası anlamlı bir şekilde gösterilemediğinden, anahtar kelimeler tam anlamıyla bir yoğun ağ kullanılarak işlenirken, sözcüklerin konumu anlamlı bir sıraya sahip olduğu için, başlık ve soyut bileşenleri işlerken farklı sinir ağları değerlendirilir. Değerlendirme sonuçları, Evrişimsel Sinir Ağının (CNN) diğer ağ türleri, Tekrarlayan ve LSTM Ağları arasında en iyi performansı sağladığını göstermektedir. Bu performans, daha düşük Ortalama Kare Hata (MSE) ve ortalama tahmin süresi ile gösterilmiştir. Ayrıca, önerilen hibrit sinir ağı, evrişimsel sinir ağını kullanarak en düşük 4.52 MSE'yi elde etmiştir. Bu sonuç, bu yöntemin bilimsel makalelerin sunulma sürecini hızlandırmakta önemli bir rol oynayabileceğini göstermektedir.

Anahtar Kelimeler: Yapay Sinir Ağları; Evrişimsel Sinir Ağları; Tekrarlayan Sinir Ağı; Uzun-Kısa Süreli Bellek(LSTM) ; Doğal Dil İşleme.

TABLE OF CONTENTS

Pages

LIST OF ABBREVIATIONS

ML : Machine Learning

ANN : Artificial Neural Network

NLP : Natural Language Processing

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

1. INTRODUCTION

With the digital revolution, most of the applications and services are being implemented online using digital information. Scientific research is not an exception from this revolution, where digital libraries are being used instead of the classical physical libraries. In digital libraries, scientific researches are being published and accessed digitally, without the need for a physical copy of the research [1]. Digitalizing these libraries have eased the access to most recent publications, which has kept researchers around the world up to date with the rapid development in different fields of study. However, with every new feature, new challenges emerge, where the existence of digital libraries has encouraged an enormous number of researchers to attempt publishing their work through these libraries [2, 3].

To assess the quality of a scientific article and decide whether it matches the requirements of a journal, experts from the field of the study that the article is involved in review these articles and provide recommendations whether to publish the article or not. This process is known as peer review and reviewers evaluate the quality based on different measures, such as the significance of the contribution, the importance of the field of the study and the presentation of the information in the article [4]. According to the enormous number of researches being submitted to the journal, these articles are being queued for a long time before submitted to the reviewers for evaluation [5, 6].

Another important feature that has been proposed by the use of digital libraries is the cross-reference. As the articles are being published digitally, the references cited in that article can be easily recognized and linked to the article. Thus, the number of times a certain article is cited in the literature can be recognized easily using the cross-reference. Articles in an important field of study and have high quality are normally cited in multiple other future articles, producing a higher impact on that field of study [7]. However, articles published earlier normally have been cited more than an article that has been published recently. Thus, the impact of an article is measured by the number of citations it gets per each year it has been published, i.e. the impact rate of an article is equal to the number of citations it gets per year. This impact rate has been widely used to represent the quality of articles, as well as the journals [8, 9].

Machine Learning (ML) attempts to use examples collected from a domain to create a model that allows computers to interact with that domain. The model is created based on the knowledge retrieved from the examples, instead of using a static set of rules created by an expert in that domain. After creating the model, predictions from the ML technique can be gained by passing the input data into the model [10, 11]. Depending on the inputs of the ML technique and the knowledge investigates in these inputs, to create the model, as well as its output, these techniques can be of unsupervised and supervised categories. For supervised techniques, the output required from the technique must be provided with each input, so that, the created model defines relations between these inputs and their assigned outputs. Regression is one of the supervised ML types that calculates a continuous output based on the values of the input data. Unlike classification, which is another supervised ML type that has discrete outputs that represent the categories of the instances in the domain, the output of regression is continuous and has no predefined limit. Thus, regression has been widely employed in applications that require continuous output, such as price, quality and time predictions [12-14].

Recently, significant attention has been attracted by Artificial Neural Networks (ANN) in different machine learning applications. Inspired by human's brains, computations in these networks are implemented around units, known as artificial neurons [15, 16]. Depending on the flow of inputs and outputs among these neurons and the external domain, ANNs have been able to process different types of data, in order to achieve different tasks. Natural Language Processing (NLP) is one of the applications that ANNs have been able to achieve outstanding performance. The complexity of the features in the text written in natural language are too complex to be detected by traditional machine learning techniques [17, 18]. However, specific types of ANNs, mainly Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), have been able to detect such features and accomplish the task required.

Both CNN and RNN can handle two-dimensional inputs but each type of these networks has a different approach in handling these inputs and detecting the features in them. In a CNN, filters are convoluted throughout the entire input in order to search for local features. By cascading these layers, more complex features can be detected, according to the ability of each layer in combining features from the previous ones. Such ability has

enabled these networks to process texts written in natural language, as words are combined into phrases and phrases into sentences in these layers to analyze the text. In RNN, the computations at a certain tuple, from the input set, is affected by the output of the recurrent layer from the previous tuple. Such topology can also consider the effect of one tuple over the next one, in which can be used to analyze the inputted text. These types of ANNs have been widely used in different NLP applications, such as sentimental analysis, classification, spelling and grammar error detection and correction [19-21].

1.1 PROBLEM DEFINITION

As a result of the digital revolution, the number of scientific researches and articles being submitted for publishing through digital libraries. To ensure the quality of the articles published by a journal, these journals require reviewing these articles by experts to asses them and advise the journal for a suitable decision. According to the relatively long time required by an expert to review an article and the enormous number of articles being submitted, the time that each submitted article is queued for reviewing is becoming behind reasonable. Moreover, according to the high rejection rates for these journals, especially prestigious ones, providing an automated scheme to asses the quality of these articles can accelerate the reviewing process significantly, by eliminating the articles assessed to be of very low quality or allowing the authors to asses and enhance their articles before submission. However, as these articles are written in natural language and have undefined length, a complex assessment scheme is required to provide accurate predictions that can be reliable enough to be considered.

1.2 THE AIM OF THE STUDY

To assess the quality of the submitted article, this study aims to employ an artificial neural network to process the texts written in natural language and provide a quality measure for each article. As such implementation is a supervised machine learning, a dataset with each article's quality measure is required. To provide the networks with actual and reliable measures, the impact rate of each article is used as its quality measure. Articles written in an important field of research and in a language suitable for the researchers are

cited more than those with fewer citations. Moreover, as the length of an article is unknown, the proposed method uses certain parts of the article to assess its quality, instead of processing the entire text. Different approaches are implemented and evaluated in order to select the most accurate model for the proposed method.

1.3 STRUCTURE OF THE THESIS

The remaining chapters of the thesis are structured as follows:

- The literature related to ANN and NLP is reviewed in Chapter Two.
- The proposed methods are described in Chapter Three.
- The results of the conducted evaluation experiments are shown in Chapter Four.
- These results are discussed and compared to other studies in Chapter Five.
- The conclusions of the study are summarized in Chapter Six.

2. LITERATURE REVIEW

Humans have gained the ability to write and read texts through thousands of years, so that, they have become capable of understanding the meaning behind the words written in a text. The ease of understanding such texts is a result of the accumulative experience in our brains to analyze the text, understand the meaning of the words and the entire text, as well as estimating the emotions of the writer while writing the text. Providing computers with such capability is not an easy task, according to the way computers are designed and implemented, which makes the semantic analysis of texts more difficult than analyzing structured data. However, certain types of artificial neural networks have shown the ability of achieving such task by considering the effect of each word as well as phrases on the overall meaning of a sentence, hence, a text. As these networks can only accept numerical inputs, certain processing is required for the text before being processed by the neural network. In this chapter, the preprocessing techniques and structure of artificial neural networks employed in the method proposed in this study are described in details.

2.1 TEXT PREPROCESSING

Several preprocessing techniques are proposed to reduce the dimensionality of a text and convert it into numerical format that can be used with ANNs. Removing any words that have no contribution toward distinguishing features in the text, such as stop words, can reduce the dimensionality of the inputs of the neural network, hence, reduce the complexity of the computations required to handle such inputs. Moreover, as words in English language have different variations, the use of a single word to represent all these variations can also reduce dimensionality, as the number of unique words in the text becomes fewer [22, 23]. However, despite the absent effect of such words and variations on the semantic meaning, they still have significant effect over the quality of the language being used in the article. Thus, these preprocessing methods are not employed in the proposed method in order to allow the recognizing the grammar and spelling errors in the text as such errors can significantly affect the overall quality of the article. However, converting the words in the text into numerical format is still mandatory as it is intended to be processed using ANNs [22, 24].

2.1.1 Text Tokenization

Each word in the text of the corpus is assigned with a unique integer number to create a dictionary that is used to convert any text into numerical value. When a certain is converted using text tokenizer, the integer value corresponding to each word in that text is used to replace the word, so that, the result of text tokenization is a series of integer values, instead of text. The number of integer values outputted from the tokenizer is equal to the number of words in the input text. To convert these values into text again, the same dictionary can be used to extract the word correspondent to every integer value in the input. Algorithm 2.1 illustrates the main steps executed by the tokenizer to convert the input text into numerical format. The output of this algorithm is a single-dimensional vector that has a length equal to the number of words in the original text correspondent to the output [25, 26].

Algorithm 2.1: Text tokenization algorithm.

Algorithm: Text Tokenization
Inputs: *Corpus*, *Text*
Output: Tokenized Text (In numerical format)

Step 1: *corpus* ← Read input corpus.
text ← Read input text to be tokenized

Step 2: *U* ← extract unique words from *corpus*

Step 3: *N* ← *1* //The first unique value to be assigned to the first word.
D ← *{}* //Initiate an empty dictionary for the corpus.
For each unique word *u* in *U*:
D ← *{D; (u, N)}* //Update the dictionary with the new word.
D ← *D+1* //Generate a new value for the next unique word.

Step 4: *O* ← *[ ]* //Initiate an empty array for the output values.
For each word *w* in *text*:
O ← *[O; D{w}]* //Append the corresponding integer value to the output.

Step 5: Return *O* //Return the tokenized text (in numerical format)

2.1.2 Vector Padding

Natural-writing text can be of any length, according to the amount of information being presented in the text and the writer's style of writing. However, according to the need of fixed-size inputs, by ANNs, it is important to convert the length of the input vectors to a constant value, so that, the same ANN can be used to process texts of any lengths. Fixing the length of the vector can be utilized by padding the vector to increase its length to the required size, or by reducing it. Reducing the size of the vector requires eliminating words from the text, by eliminating the tokenized values. However, as these words can be of significant importance to the evaluation process and their removal could drop such information from being processed by the ANN, padding is used to increase the length of the vectors less than the length of the longest text in the corpus. To avoid confusion, the value used for the padding does not exist in the dictionary created in the tokenization process, so that, the ANN can recognize the difference between the padding and actual words from the corpus [27, 28]. Algorithm 2.2 shows the process executed to pad vectors outputted from the tokenizer.

Algorithm 2.2: Padding tokenized vectors algorithm.

Algorithm: Vectors Padding. **Inputs:** Vectors of different lengths. **Output:** Padded vectors with fixed length.
Step 1: V [,]← Read input vectors. **Step 2:** L ← Length of the longest vector in V. **Step 3:** C ← number of vectors in V. //Number of texts in the input. $O[C, L] \leftarrow 0$ //Initiate output array with fixed dimensions and zero values. For $x = 1$ *to* C: s ← size of vector $V[x]$ //Get the size of the original vector. $O[x,:s] \leftarrow V[x]$ //Place the values from the vector in the beginning of the corresponding output vector. **Step 4:** Return O //Return the fixed-size vectors array.

2.2 ARTIFICIAL NEURAL NETWORKS

Inspired from humans' brains, computations in ANNs are implemented in units, known as artificial neurons, distributed over the network in layers. The inputs of a certain neuron can be collected from the external domain or from the outputs of the previous layer's neurons. To calculate the output of a neuron, all collected inputs are weighted, by multiplying each of them with a certain value assigned per each input and summed, before being passed through a nonlinear function, known as activation function, as shown in Figure 2.1. This nonlinearity provides more flexible output that has the ability to detect more complex features. Nevertheless, additional value can be added to the inputs of a neuron to provide bias to the computations, when needed, known as the bias [29, 30].

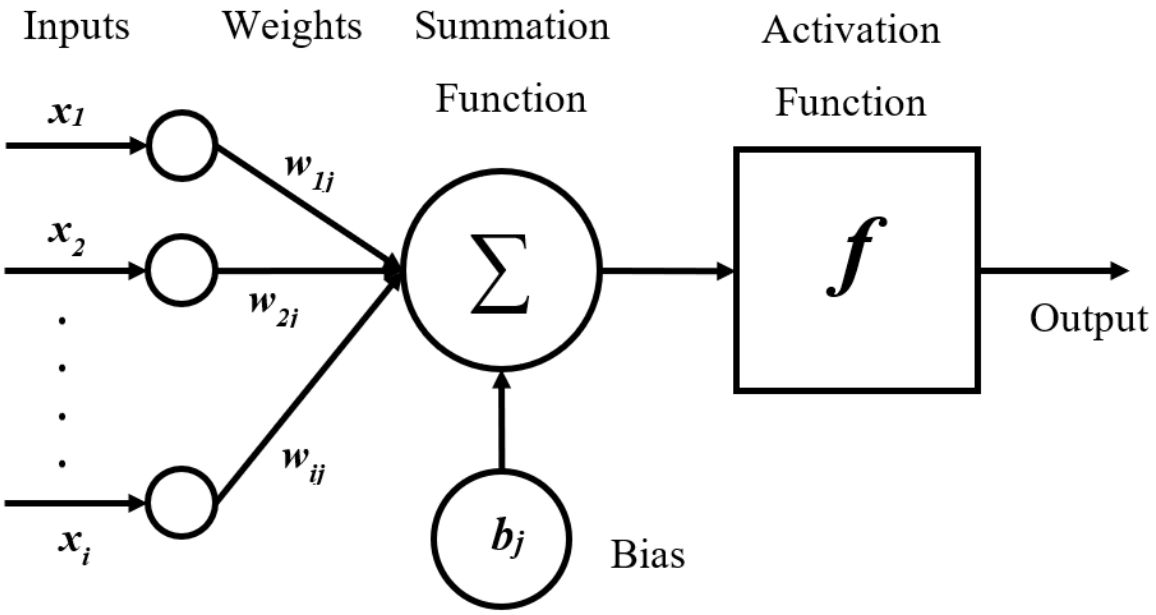

Figure 2.1: Illustration of the computations inside an artificial neuron [29].

Regardless of the type of the ANN, each of these networks has two types of computations, one executed from the input to the output direction, known as the forward pass, while the other is executed in the opposite direction, known as the reverse pass [31]. The forward pass is used to calculate the output of the network, based on its inputs, by calculating the output of each layer and use in the computations executed in the second one. In the reverse pass, the weights' values are updated through gradient descent. By measuring

the deviation between the output of the ANN, from the forward pass, and the intended output values, from the dataset, the derivatives of the output to the weights are calculated. Gradient descent is used to recognize the position weights' value must be updated to reduce that error, which is to the negative of the gradient decent at that position. Such update allows the neural network to produce the intended output from the inputted values, hence, achieve the required task. By repeating this process for several iterations, the loss between the output from the forward pass and the intended output is reduced using backpropagation, which improves the performance of the neural network, until the minimum loss is reached [32, 33].

2.2.1 Convolutional Neural Networks

CNNs contain convolutional layers, which consists of two-dimensional filters that are convoluted throughout the input of each neuron. Mathematically, the filter is actually the weight values of that neuron, which enable the neuron to detect local two-dimensional patterns in the input. The sizes of the filters in a convolutional layer is constant and patterns in the input can be detected within the size of the filter. However, by going deeper into the neural network, i.e. layers farther from the input layer, each filter detects patterns defined by the patterns detected by the previous layer's filters. This enables the CNN to combine the recognized patterns and detect more complex features. Although the output of a neuron in a convolutional layer can have different dimensions from its input, the number of dimensions is similar to that in the input, i.e. a neuron processing a two-dimensional input outputs a two-dimensional array [34, 35].

During convolution, the number of values that the filter moves per each step is defined as the strides, which can have different values for the horizontal and vertical movements. All the values within the filter are multiplied with their corresponding weights and processed in the neuron, which arranges its outputs according to the arrangement received during the convolutions of its filters. Skipping more than one value per each convolution can cause the loss of detecting important patterns, which can negatively affect the performance of the CNN, despite the reduction in the size of the neuron's output, which can simplify the computations in following layers. To reduce the size of the output from a

neuron without losing important information, pooling layers can be placed after a convolutional layer [36].

A pooling layer also consists of filters that are convoluted throughout its input, which is the output of the neuron. However, these filters have a different approach to process the input values, as they are not forwarded to a neuron and has no weights. Despite the existence of different types of pooling layers, Max-Pooling layer is one of the widely used pooling layers that are used to reduce the size of the processed data without losing important information. As shown in Figure 2.2, the filter in a max-pooling layer searches for the maximum value within its dimensions, and outputs that value to represent that region. By selecting the highest value, the most important feature in that region is selected, so that, it is less likely to lose important information as in increasing the strides of the filter in the convolutional layer [36].

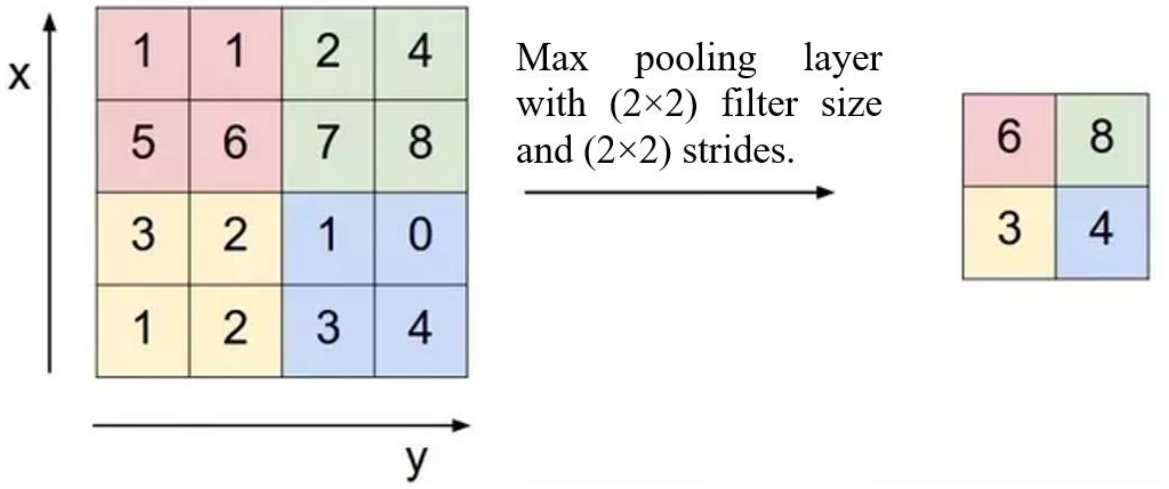

Figure 2.2: Output of Max-Pooling filter.

According to the ability of CNNs to consider the position of an input, in addition to its value, these networks are being widely employed in NLP. For example, such network can recognize that the phrase *"does not exist"* is equivalent to the word *"absent"* in a sentence, so that, the effect of these two neurons can be similar with respect to the output of the neural network. Moreover, when the output required from the neural network is not two-dimensional, which is the case in most applications, the output of the last convolutional layer can be flattened and fully connected to another one-dimensional layer. Depending on

the complexity of the features in the input, more layers can be added to the neural network before the output layer [37, 38].

2.2.2 Recurrent Neural Network

Similar to CNNs, recurrent neural networks can handle two-dimensional inputs and output a single value per each set of inputs. However, the approach RNNs use to process these inputs is different, where the output from a previous input tuple is weighted and appended to the inputs collected from the previous layer, or the external domain. As shown in Figure 2.3, suppose a weight value *f* is used to adjust the value of the output from the tuple previous to the current tuple positioned at *t*. During the computations of the output of the neuron at *t*, the output *h* from *t-1* is included after being weighted using *f*. The output at this *t* tuple is also weighted using *f* and included with the inputs *x* of the next tuple at *t+1*. This process is repeated until all the tuples in the input set are processed [39, 40].

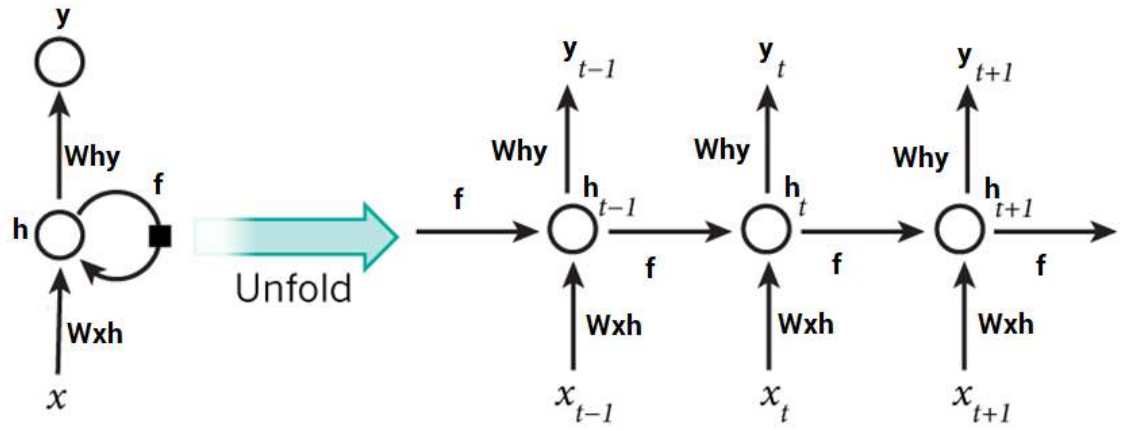

Figure 2.3: Computations in an RNN neuron.

According to the ability of RNN's to include outputs calculated from previous tuples in the computations of the current one, this type of neural networks is widely used in timeseries and NLP applications. A phrase can be analyzed according to the effect of each word in that phrase and its position. For instance, the output of processing a negative word, such as *not*, can be combined with the inputs of the next word, so that, the meaning of that word can be inverted. Moreover, errors can be detected by recognizing wrong

combinations, when a word following another is in wrong formation, depending on the definition of the suitable form in the grammar [41, 42].

2.2.3 Long- Short-Term Memory

As illustrated in the previous section, the effect of a certain output from the neuron is relative to the position of the tuple being inputted to the network, with respect to the one being processed in this instance. At instance *t*, the output from *t-1* has more influence on the current output than that from *t-2*. However, in many applications including NLP, such behavior can be of significant importance in certain conditions, and of negative influence in other. Thus, a more complicated type of RNNs is being used in these applications, where the influence of a certain output is adjusted according to its importance in the current computations, rather than its position in the series [43].

To achieve such a task, LSTM networks use gates to control the flow of the values between the input and the output. Each gate is controlled using a separate network that accepts inputs from certain position. As shown in Figure 2.4, net_c is the input network that receives the values from the external domain and calculates the outputs depending on its weights. Another network net_{in} receives a copy of these inputs in order to control the gate that defines the flow of the output from net_c, through the input gate value y^{in}. The effect of the previous output is adjusted using the forget gate values y^{ϕ}, which is controlled using net_{ϕ}. This output S_c is squashed using an activation function before being adjusted using the values y^{out} acquired from the output gate, which is controlled using net_{out} that calculates the values of the gate using the outputs collected from the previous time instance. As each gate is controlled using a different neural network, the weights of each neural network are updated during the training of the networks, so that, the appropriate decision is made based on the input values of the current time instance and the outputs collected from the previous ones [44].

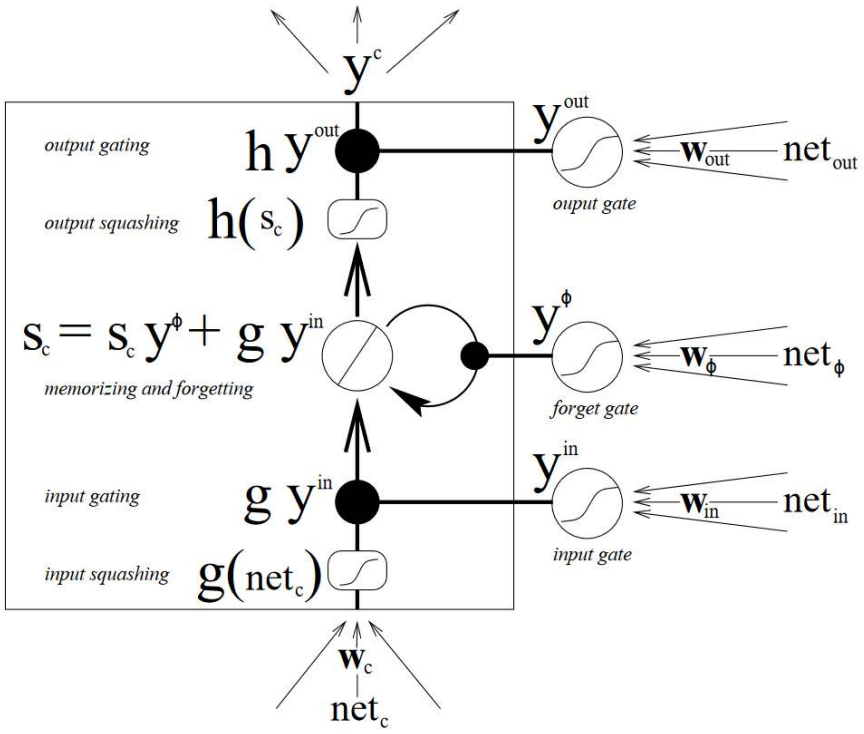

Figure 2.4: Illustration of the data flow in an LSTM neural network [44].

2.3 PERFORMANCE EVALUATION

As ML techniques learn by examples, the accuracy of the extracted knowledge, from these examples, can have significant influence on the performance of the neural network. Moreover, as illustrated earlier, ANNs employ different approaches to extract the model that can be used to process future inputs, which produces different models. To select the approach that is most suitable for the required task, the performance of each of them must be evaluated. This evaluation is conducted by using the same dataset collected from the domain, which is also used for training. However, including the instances in the evaluation set in the training set eases the task for the ML technique, as the patterns in these inputs are already included in the extracted model [45]. To avoid such behavior, a set of inputs is excluded from the training dataset, so that, their features and patterns are not included in the model. The selection of the instances for the evaluation can result in biased measures, as the selected instances can be more suitable for models created by certain techniques than other [46]. Thus, a different evaluation technique is used that allows the use of the entire

dataset for evaluation to produce unbiased measure, which is known as k-fold cross-validation.

The dataset collected from the domain is split into *k* bins, in k-fold cross-validation. Then, by iterating through these bins for *k* times, the ML technique is trained using *k-1* bins, while the remaining bin is used for the evaluation. As the number of iterations is equal to the number of bins, cross-validation ensures to use each of the bins once for evaluation, so that, by the end of the iterations the entire dataset is used for evaluation. As, in such approach, the entire dataset is used, the bias in the evaluation is minimized. However, studies show that cross-validation is not bias-free but it still produces less biased evaluation than randomly splitting the dataset into training and testing sets [47-49].

Per each iteration, a performance measure is calculated to illustrate the quality of the outputs collected form the model at that iteration, by comparing them to actual values collected from the domain. For regression problems, Mean Squared Error (MSE) is widely used to measure the quality of the predictions. As shown in Equation 2.1, the error between the predicted p_i and actual a_i values is squared and averaged for all the I instances in the testing dataset. As the value of the MSE increases when the difference between the predictions and actual values increase, the performance of the technique is considered better when its MSE is lower [50].

$$MSE = \frac{\sum_{i=0}^{I}(p_i - a_i)^2}{I} \tag{2.1}$$

3. METHODOLOGY

As the significance of a scientific article can be measured based on the field of the study it is involved in and the quality of the writing of the article, the proposed method three of the main components of an article, the title, keywords and abstract. The analysis of the title and keywords can provide good knowledge about the field that the study is involved in. Additionally, the analysis of the abstract can also provide knowledge about the quality of the writing and the existence of grammar and spelling error, which significantly reduce the quality of the article. This selection also avoids the complex analysis of the entire article, which can be extremely long and adds no further knowledge to the assessment.

Each of the selected components is trained and processed using a designated neural network. As each keyword in an article may consist of one or more words, these words are fused together, so that, a certain keyword is maintained consistently throughout the entire corpus. Moreover, as the position of a keyword, with respect to another, has no valuable information about the quality of the article, i.e. the order of these keywords is irrelevant, the neural networks implemented for the keywords analysis is a feed-forward neural network. Each neuron in a certain layer collects its inputs from the outputs of the previous layer, or the external world for the input layer.

3.1 PREPROCESSING

Three text corpora are generated from the articles, one per each of the titles, keywords and abstracts. These text corpora are then used to tokenize each of these components per each of the articles, so that, the integer value assigned for a certain word is maintained consistent throughout all the articles. This approach allows the extraction of more useful patterns that can be used in the assessing the quality of the writing, where important and undesired phrases can be detected in all articles. Moreover, as punctuations in the abstract can have a significant influence on assessing the writing quality and as these punctuations are normally placed adjacent to the previous word without a space, a space is placed before every punctuation, so that, the position of the punctuation mark can be recognized when the

text is split by spaces. Algorithm 3.1 summarizes the preprocessing procedure conducted using the articles. Finally, the text of each of these components is tokenized and padded to created a fixed-size numerical vectors for these components.

Algorithm 3.1: Preprocessing of the articles before being assessed using the proposed method.

Algorithm: Articles Preprocessing
Inputs: Articles
Output: Fixed-size vectors for the title, keywords and abstract.

Step 1: *A* ← Read articles.
T ← [] //Initiate an empty array for titles.
K ← [] //Initiate an empty array for keywords.
B ← [] //Initiate an empty array for abstracts.

Step 2: For each article *a* in A:
T ← *[T; title of a]* //Append the title to the titles array
K ← *[T; keywords of a]* //Append the title to the keywords array
B ← *[T; abstract of a]* //Append the title to the abstracts array

Step 3: For each keyword *k* in *K*: //For each keyword in the keywords
k ← remove white spaces from *k*. //The words in each keyword are fused in a single word.
For each word *b* in *B*: //For each word in the abstracts
If last character in *b* is a punctuation mark:
b ← *b[:-1] + ' ' + b[-1]* //Add a white space between the word and the punctuation mark.

Step 4: *TT* ← *Tokenize (T)* //Tokenize the titles using Algorithm 2.1
TK ← *Tokenize (K)* //Tokenize the keywords using Algorithm 2.1
TB ← *Tokenize (B)* //Tokenize the abstract using Algorithm 2.1

Step 5: *OT* ← *pad(TT)* *//Pad the tokenized vectors using Algorithm 2.2*
OK ← *pad(TK)* *//Pad the tokenized vectors using Algorithm 2.2*
OB ← *pad(TB)* *//Pad the tokenized vectors using Algorithm 2.2*

Step 6: Return *OT, OK, OB* *//Return the padded tokenized vectors.*

3.2 ARTICLES' QUALITY ASSESSMENT

Besides the keywords of an article, as their order does not have important information, the title and the abstract can be processed using CNN, Simple-RNN of LSTM neural networks, as the positions of their words have important information toward assessing their quality. Two approaches are proposed in this study to measure the overall quality of the article. The first approach attempts to predict the quality from each of these components, solely, then use a statistical method to fuse the three measure into one. The other approach uses a single hybrid neural network that accepts all the inputs from the article and produces a single output directly.

3.2.1 The Statistical Fusion Approach

As shown in Figure 3.1, a separate neural network is used per each of the components extracted from an article, i.e. one neural network per each of the title, keywords and abstract. Each of these networks is trained using the correspondent component as an input, while the required output is set to be the overall quality measure of the article. Using the three quality measures, an overall value is calculated using a statistical function, such as the average or median function. As all networks are trained to produce the same value, the use of the statistical function can reduce the error if produced by any of the neural networks, hence, improve the performance of the proposed method.

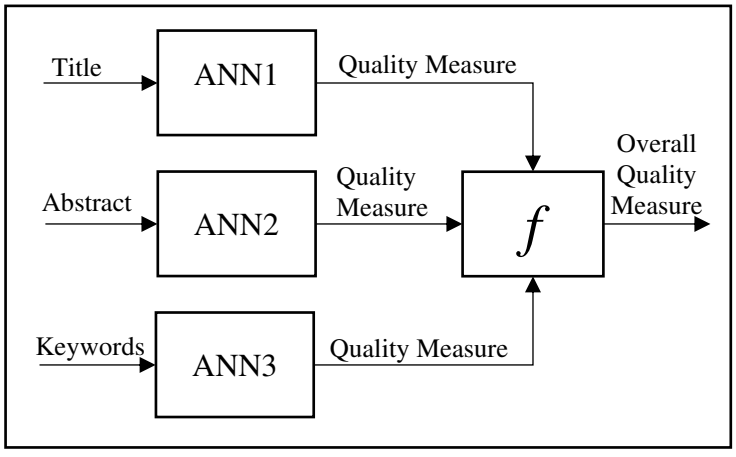

Figure 3.1: Illustration of the statistical-based article's quality assessment approach.

3.2.2 The ML Fusion Approach

In this approach, a single hybrid neural network is implemented to directly provide the overall quality measure of the article, instead of calculating three of them separately. Out of each neural network, an eight-value vector is forwarded to a 64-neuron layer in a fully-connected network. This layer is followed by four hidden layers, with 128, 64, 32, 16 and 8 neurons, respectively. Finally, an output layer with no activation function, as it is a regression problem, is used to output the overall quality measure of the inputted article. Figure 3.2 summarizes the structure of the hybrid neural network implemented for this purpose.

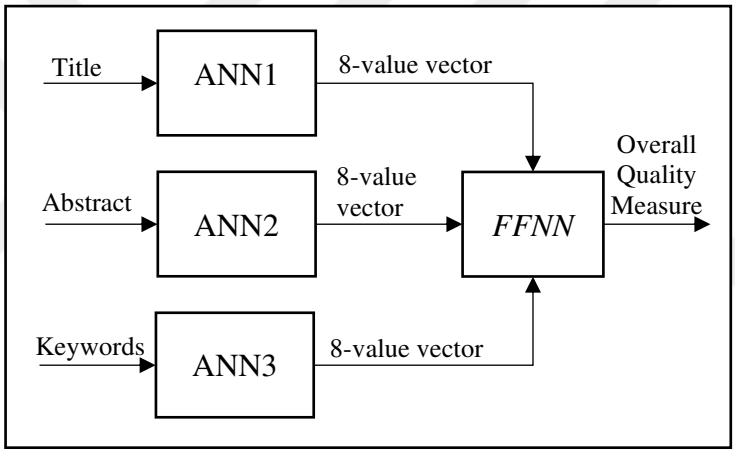

Figure 3.2: Structure of the proposed hybrid neural network for article's quality assessment.

All the inputs generated from the three components of the article are placed in a single vector, as the implemented hybrid neural network is trained to process all these inputs simultaneously. However, up to the last layer in each component's neural network, no interconnection is implemented between these networks. Each subnetwork processes its input and provides an 8-value vector that represents it. Then, the FFNN combines these vectors in order to output the corresponding quality measure. However, as the neural network is implemented as a single hybrid network, the weight's update procedure relies on the difference between the outputted quality measure and the required one.

4. EXPERIMENTAL RESULTS

Each of the components used for the evaluation is the tokenized, separately, and padded, so that, all the texts per each component have the same length of numerical values. All experiments are conducted using a Windows computer with Intel Core i7-7700HQ processor and 16GB of memory, with a Graphics Processing Unit (GPU) with 8GB of memory. The GPU is used to accelerate the mathematical operations required by the artificial neural network, according to the ability of GPU to parallelize matrices operations. All evaluations are performed using 5-fold cross-validation, to avoid any biased results toward any of the implemented neural networks, where the methods are implemented using Python programming language.

4.1 SUMMARY OF THE DATASETS

To evaluate the performance of the proposed method, two UCI [51] datasets, collected from the Association for the Advancement of Artificial Intelligence (AAAI) journal. The datasets contain the title, keywords and abstract of the articles accepted by that journal in 2013 and 2014, a dataset per each year. However, these datasets do not contain the number of citations each article has earned, which are used as the quality measured for these articles to train and evaluate the performance of the proposed methods. The number of citations per each article in the dataset is collected, one by one, from Google Scholar indexes search engine. The collected numbers of citations are then divided by the number of years the article has been published for and used as a measure of the quality of that article. Some of the articles in these datasets are not available in the Google Scholar respiratory, as shown in Table 4.1. These datasets are combined into a single dataset for the training and evaluation of the proposed method. Two of the articles in the resulting dataset have citation rates higher than 100/year. These numbers are reduced to 100, in order to avoid overfitting the neural networks. The histogram shown in Figure 4.1 represents the frequency of each impact rate in the collected dataset, which shows that most of the articles in the dataset have an impact rate between zero to 20 citation per year.

Table 4.1: Summary of the datasets used for evaluation.

Dataset	No. of Articles	Found Articles	Missing Article
AAAI 2013	150	138	12
AAAI 2014	399	381	18
Total	**549**	**519**	**30**

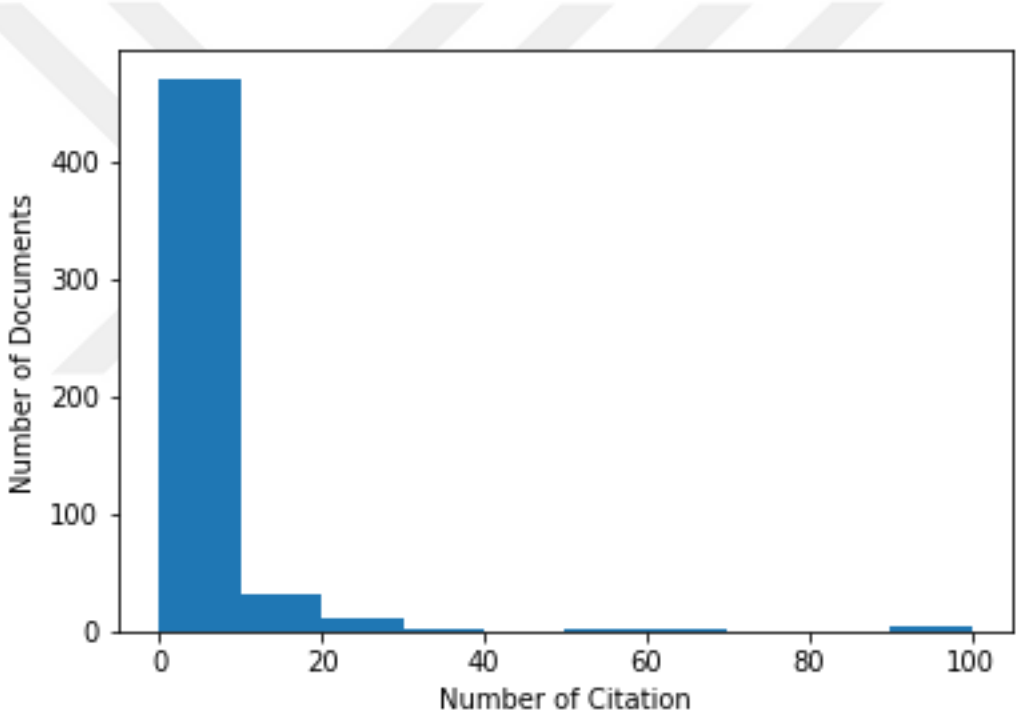

Figure 4.1: Histogram of the impact factors for the articles in the collected dataset.

As shown in Table 4.1, the resulting dataset that is used for the training and evaluation, using the 5-fold cross-validation, consists of 519 articles, out of the total 549, as 30 of the articles in these datasets are not found in Google Scholar index. Three text corpora are generated from these articles, one per each of the title, keywords and abstract components of these articles. The characteristics of these corpora are shown in Table 4.2.

Table 4.2: Characteristics of the generated text corpora for the articles' components.

Text Corpus	No. of Unique Words	Maximum Length
Title	1670	16
Keywords	1587	16
Abstract	10287	343

4.2 PERFORMANCE OF THE STATISTICAL FUSION APPROACH

In this experiment, neural networks are implemented and trained to measure the quality of each article based on each of the components. CNN, Simple-RNN and LSTM networks are implemented and evaluated for the titles and abstracts of the articles, while a feed-forward neural network is used for the keywords as the order to these words has no significant knowledge. First, the performance of each of these neural networks is illustrated. Then, the overall performance by fusing the results, using the average and median function, is also illustrated.

4.2.1 Quality Assessment Using Articles' Titles

Three neural networks, with six hidden layers, in addition to the input and output layers. Four of the hidden layers are special layers, i.e. CNN, RNN or LSTM layers, while the other two are fully-connected dense layers that accept the flattened output of the last special layer. As shown in Table 4.2, the maximum length of the titles is 16, which indicate that the width of the input vectors is 16 as well, after padding shorter titles. Thus, the input layers of the implemented neural networks consist of 16 neurons, while the output layer has a single neuron as a single value is required. Table 4.3 summarizes the MSE measured for each type of neural networks, using the title as input.

Table 4.3: Performance evaluation summary of the title-based articles' quality assessment.

Neural Network	MSE	Prediction Time (*ms*)
CNN	12.06	2.12
Simple-RNN	24.73	3.71
LSTM	18.15	3.97

As illustrated in Figure 4.2, the CNN has been able to achieve the lowest MSE, i.e. the best performance, when predicting the quality of the articles based on their titles. Moreover, the figure also shows that the CNN has also consumed the lowest average time per prediction, which illustrates that the computations in these networks is less complex than in the RNN and LSTM networks. Moreover, a significant difference in MSE can be noticed between the LSTM and simple-RNN networks, with only slightly more complexity indicated by the longer execution time. These differences illustrate the importance of the gates in the LSTM that controls the flow of the data in the network.

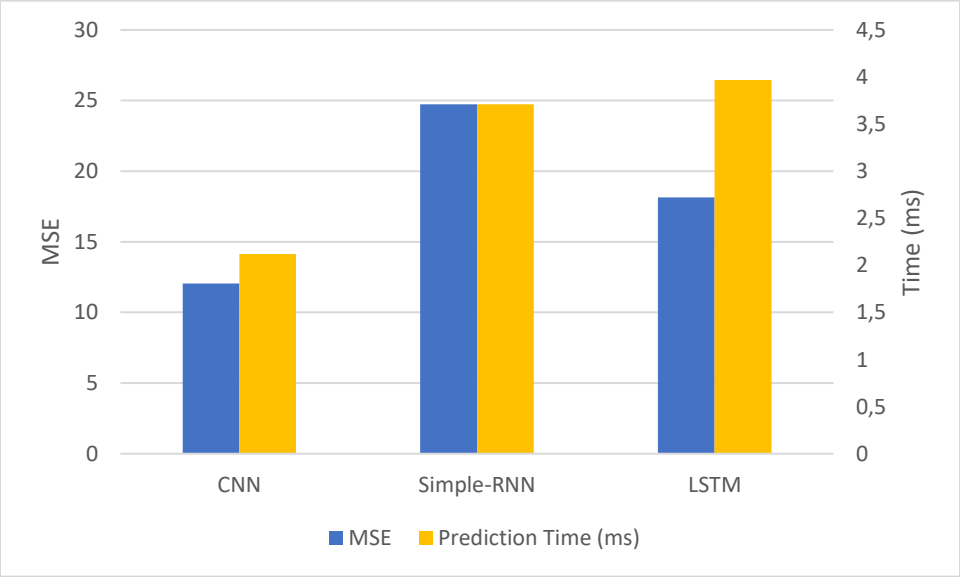

Figure 4.2: Illustration of the title-based articles' quality assessment performance.

As the CNN has the lowest MSE among the evaluated types of neural networks when the titles of the articles are used to measure their performance, the actual and predicted

impact rates are illustrated in Figure 4.3. This figure shows that the highest errors occur with the articles that have high impact, as the frequency of such articles is low in the dataset, which makes the recognition of distinctive features in the relatively low-size title difficult.

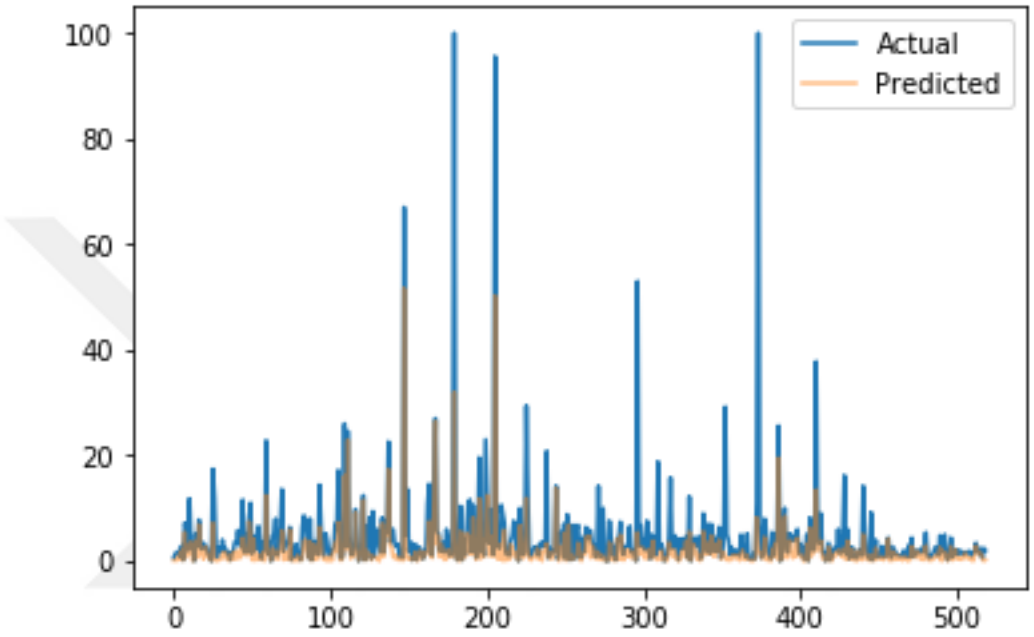

Figure 4.3: Actual and predicted impact rates of the title-based evaluated articles using CNN.

4.2.2 Quality Assessment Using Articles' Keywords

As the order of the keywords does not have a significant effect on the meaning of these keywords, where moving a keyword before or after another does not change the meaning of these keywords, a feed-forward neural network is used to process these keywords. Keywords that have more than one word in it are joint together and tokenized as a single word. As the longest keywords in the articles has 16 keywords, the implemented neural network has 16 neurons in the input layer, followed by 64, 32, 8 and 1 neurons for the hidden and output layers, respectively. The MSE of the predictions provided by this neural network is 28.86, with an average prediction time of 1.06ms per prediction. The predicted and actual impact rated for this experiment are shown in Figure 4.4. These results show that the keywords of an article do not provide enough information for the neural network to

assess its quality, especially that these words do not reflect any of the writing quality, as they are not sentence-ordered.

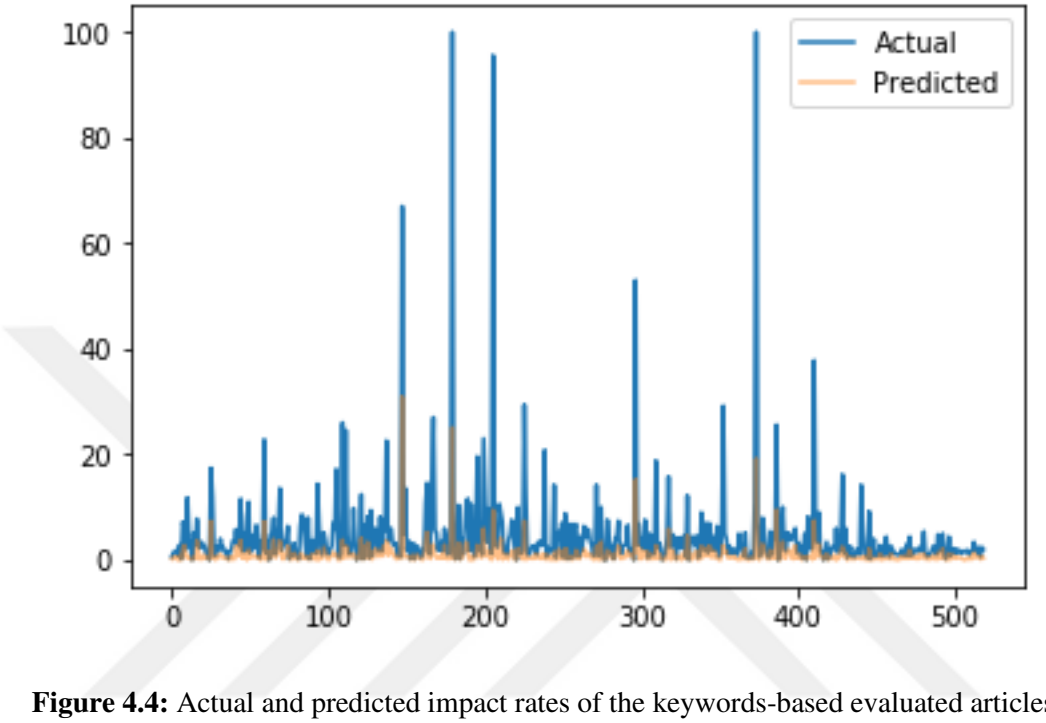

Figure 4.4: Actual and predicted impact rates of the keywords-based evaluated articles.

4.2.3 Quality Assessment Using Articles' Abstract

In this experiment, the abstracts of the articles are used for the evaluation, using the three types of neural networks, CNN, simple-RNN and LSTM. As the vectors are padded to 363 length, when a fewer number of words exist in their abstracts, the input layers of the implemented networks consist of 363 neurons. These layers are followed by 512, 256, 128, 64, 64, 32 and 1 neurons for the hidden and output layers. Four of these hidden layers are special layers, i.e. CNN, RNN or LSTM, while the other two are fully-connected dense layers. The evaluation results, described by the MSE and average prediction time, are summarized in Table 4.4.

Table 4.4: Performance evaluation summary of the abstract-based articles' quality assessment.

Neural Network	**MSE**	**Prediction Time (*ms*)**
CNN	7.37	163.71

Simple-RNN	16.25	218.52
LSTM	14.31	273.69

As shown in Figure 4.5, the best predictions have also been produced by the CNN but, surprisingly, the difference in the performance among the evaluated methods show that the performance of the LSTM neural network is more similar to the simple-RNN than the CNN. The increased number of instances per each input is expected to improve the performance of the LSTM, compared to the first experiment, as more knowledge is extracted, especially about the positioning of the words, but the results show that no such case exists.

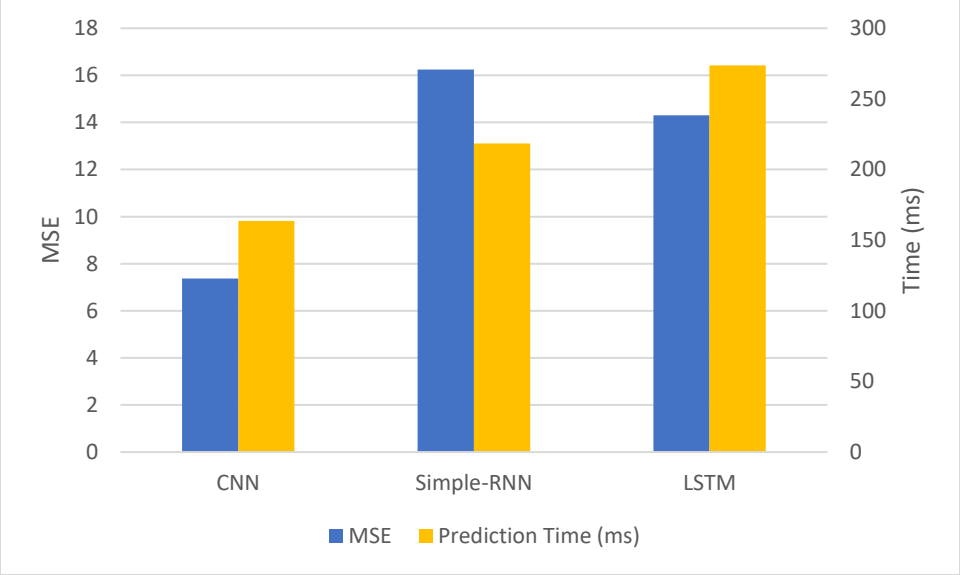

Figure 4.5: Illustration of the abstract-based articles' quality assessment performance.

The illustration of the predicted and actual values, shown in Figure 4.6, show that the more information in the abstract has been able to improve the predictions of the neural network, where more distinctive patterns can be detected to assess the quality of the article. However, the limited number of articles with high impact rate has also limited the performance of the neural network, as in cross-validation some of the training sets may not include such articles at all, which limits the ability of the neural networks to output such values.

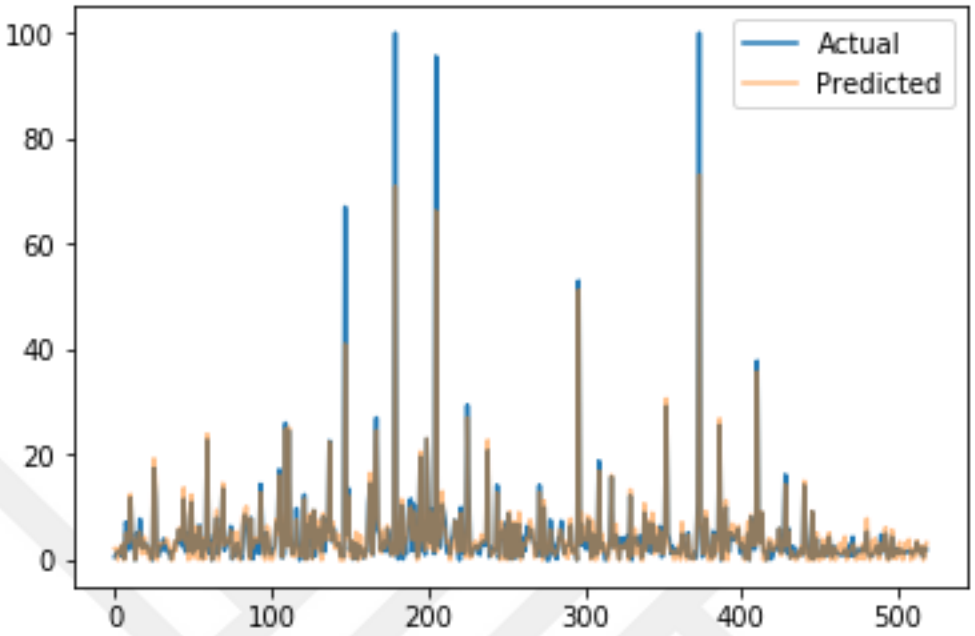

Figure 4.6: Actual and predicted impact rates of the abstract-based evaluated articles using CNN.

4.2.4 Overall Quality Assessment

As the one-dimensional convolutional neural networks have achieved the highest performance, i.e. least MSE values, in both title and abstract processing, these networks are used alongside with the feed-forward fully connected network, used for the keywords. The resulting model has an overall MSE of 16.10 when the average function is used and 8.47 when the median function is used. Despite the ability of the median function to eliminate extreme noise from the values, it has not been able to improve the performance of the quality assessment, where the MSE of the abstract-based quality assessment is 7.37, which is lower than the MSE of the fused quality measures. These MSE values illustrate the importance of providing more information to the neural network in order to allow more accurate predictions, as the abstracts of the articles have the highest number of words among the extracted components. The output of the median function, as it has the lowest MSE, are illustrated in Figure 4.7 alongside with the actual rates, which show that most of the outputted values are selected from the output of the abstract-processing neural network.

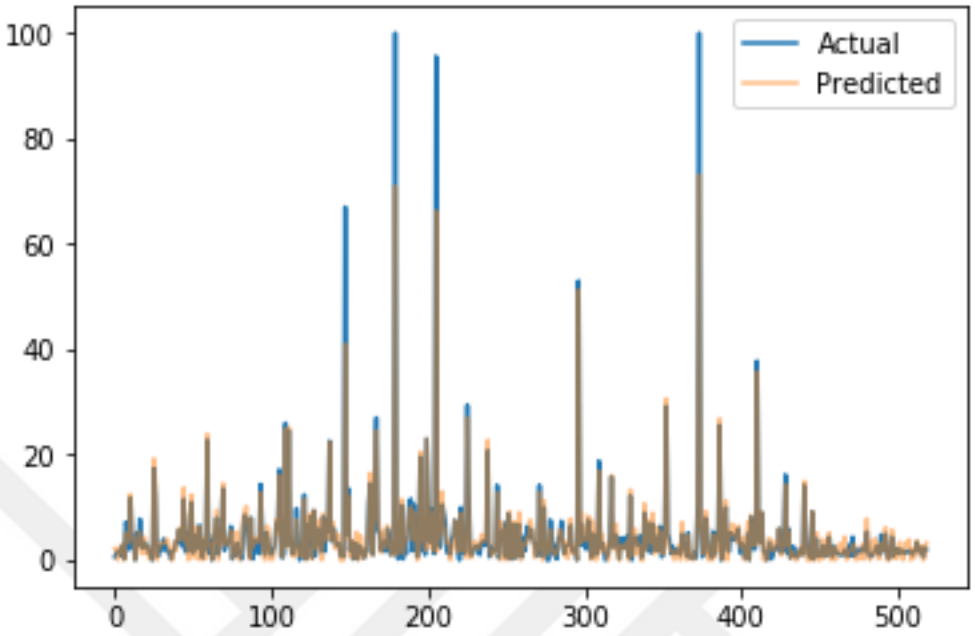

Figure 4.7: Actual impact rates and the fused output using the median function.

4.3 PERFORMANCE OF THE HYBRID NEURAL NETWORK

The performance of the proposed hybrid neural network is evaluated in this experiment. The neural networks described in the previous experiments, per each component, are implemented for this experiment, except the output layer. Thus, the output of each neural network is an eight-value vector, as the last hidden layers of these networks contain eight neurons. The 32 values collected from these networks are forwarded to three fully-connected hidden layers with 128, 64 and 32 neurons, respectively. As a single output value is required from the neural network, a single neuron is placed in the output layer. The evaluation results of this hybrid neural network are shown in Table 4.5.

Table 4.5: Performance evaluation summary of the proposed hybrid neural network.

Neural Network	MSE	Prediction Time
CNN	4.52	187.19
Simple-RNN	12.91	331.6
LSTM	7.66	409.48

Similar to the previous experiments, the use of convolutional layers to process the input values has been able to achieve the best performance, with lowest MSE and shortest prediction time, as shown in Figure 4.8. The performance of the LSTM neural network, in this topology, is more similar to the CNN than the simple-RNN, regarding the MSE, but requires the longest prediction time, according to the complex computations imposed by the additional neural networks that control the gates in the network.

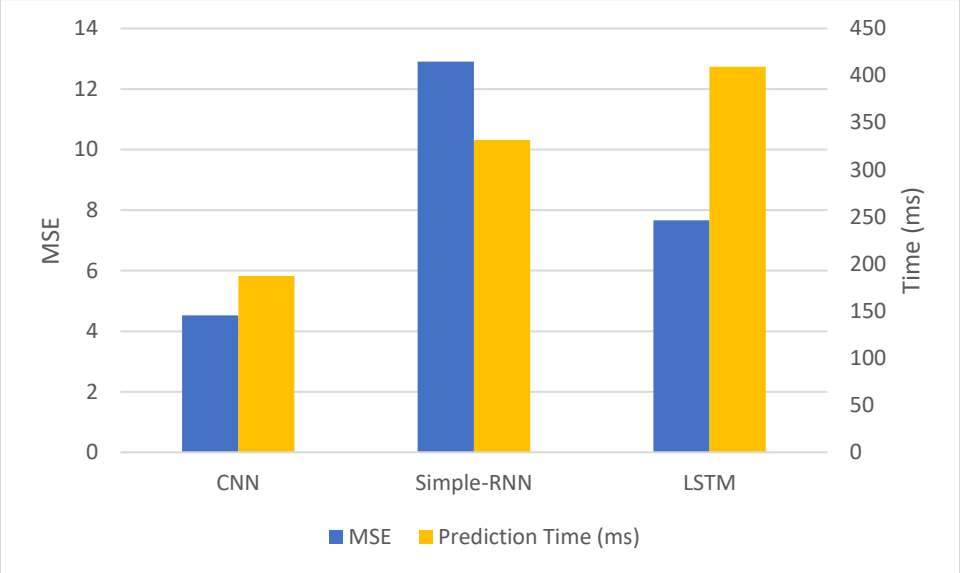

Figure 4.8: Illustration of the hybrid articles' quality assessment neural network performance.

Figure 2.9 shows the difference between the impact rates predicted by this hybrid neural network, as quality measure, and the actual one from the dataset. The figure shows that most of the errors occur with the articles that have high impact rates, according to the lack of enough samples with such rates to allow the neural network to recognize them better. However, in the lower range of impact rates, where the frequency of such articles is relatively higher, the hybrid method has been able to provide accurate predictions.

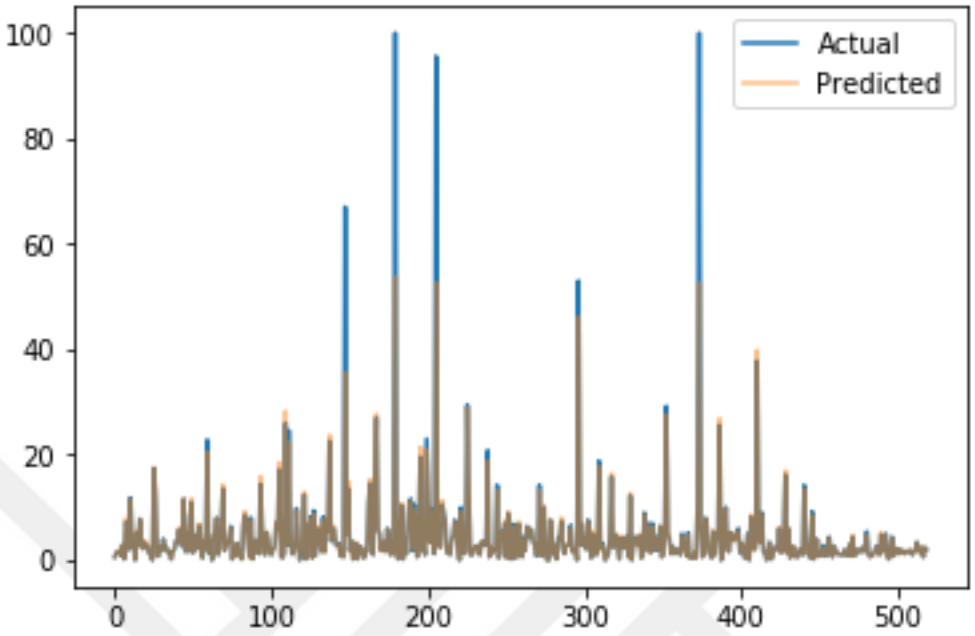

Figure 4.9: Actual and predicted impact rates of the proposed hybrid neural network with convolutional layers.

5. DISCUSSION

The comparison of the performances of the evaluated types of the neural networks, shown in Table 5.1, show that the CNN has achieved the lowest average MSE, which indicates more accurate predictions as also shown in Figure 5.1. Despite the more suitable hierarchy of the LSTM for text processing, CNNs have been noticed to produce better results in different comparisons, which has also encouraged embedding convolutional layers with LSTM models for text analysis [52-54]. The capability of considering the position of the pattern or feature of the CNN is the main reason behind such superiority. However, the LSTM neural networks have shown better performance than the simple-RNNs, in means of predictions accuracy, according to the way features are extracted from NLP texts, where older features can have more influence on the computations in the current position.

Table 5.1: Comparison of the MSE values for the evaluated neural networks.

	MSE		
	CNN	**Simple-RNN**	**LSTM**
Title	12.06	24.73	18.15
Abstract	7.37	16.25	14.31
Hybrid	4.52	12.91	7.66
Average	7.98	17.96	13.37

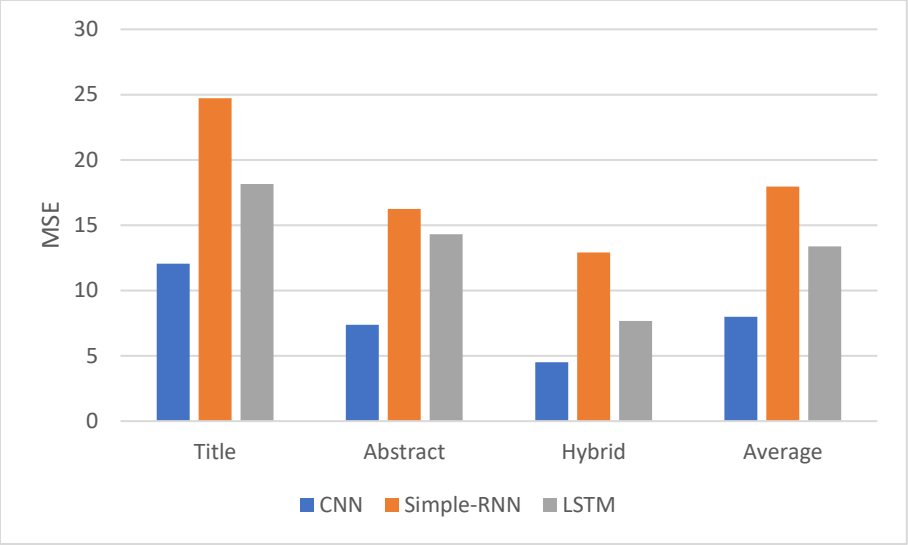

Figure 5.1: Illustration of the performances of the evaluated neural networks.

Moreover, the CNN has also shown the least average prediction time, as shown in Table 5.2 and illustrated in Figure 5.2. Such lower time consumption indicates that the computations implemented in the CNN are less complex than those in the RNN and LSTM networks. The LSTM neural network has consumed the highest average prediction time, which is a result of the existence of multiple neural networks that control the different gates that manage the flow of the data in the network [55-57]. The comparison also shows that the average prediction time is highly dependable on the complexity of the neural network, which is implemented based on the size of the inputs, where processing the title with a maximum size of 16 requires simpler network, hence, less prediction time.

Table 5.2: Comparison of the average prediction time for the evaluated neural networks.

	Average Prediction Time (*ms*)		
	CNN	**Simple-RNN**	**LSTM**
Title	2.12	3.71	3.97
Abstract	163.71	218.52	273.69
Hybrid	187.19	331.6	409.48
Average	117.67	184.61	229.05

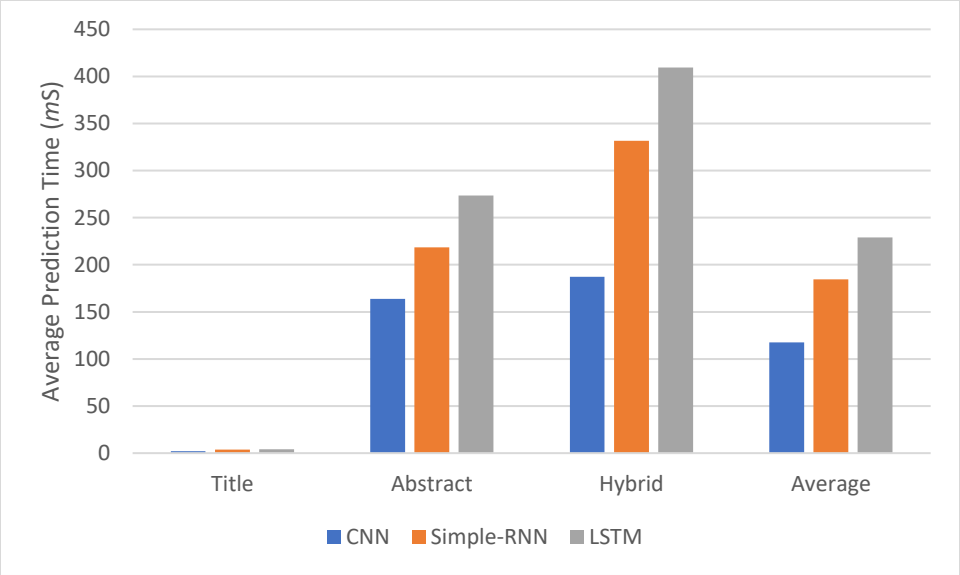

Figure 5.2: Illustration of the average prediction time of the evaluated neural networks.

Despite the good performance of the neural networks, the information that the inputs hold for the network to extract is an extremely important factor that can affect the performance of the neural network. Table 5.3 compares the size of the input to the MSE of the best prediction provided by the neural networks, which are illustrated visually in Figure 5.3. The comparison shows that increasing the size of the input significantly improve the performance of the neural network, as more knowledge is fed to the network. Moreover, despite the similar input size in the title and keywords processing, the title has more information to the neural network, presented by the positioning of the words, which can be also used to represent the writing quality of the article. Providing more data, as well as both the field of the study of the article and the writing quality, by the abstract has been able to significantly improve the performance. Eventually, providing all the available data to the hybrid neural network in order to allow it to detect the features required to assess the article has produced the best predictions. This goes along with evaluation results from earlier studies, which show that the performance of the neural network can be significantly improved by increasing the amount of data being fed to the network but the complexity of the network is increased to handle larger inputs [58, 59].

Table 5.3: Comparison of the input size to the MSE of the neural networks.

	Input Size	MSE
Title	16	12.06
Keywords	16	28.86
Abstract	363	7.37
Hybrid	395	4.52

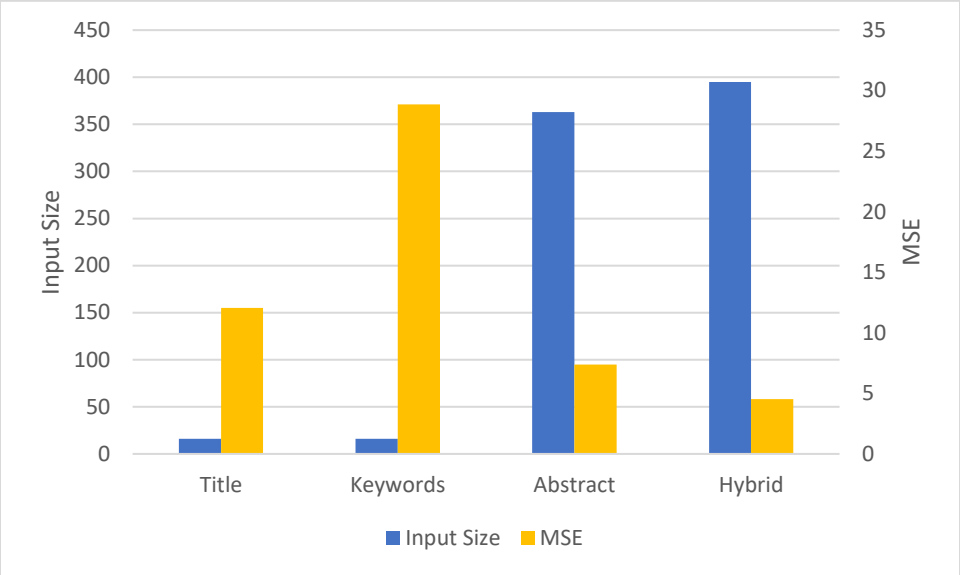

Figure 5.3: Illustration of the input size to the MSE of the neural networks.

6. CONCLUSION

The number of scientific articles being submitted for publishing every day is increasing rapidly, according to the ease of access imposed by the rapid development of internet and digital libraries. Journals are assisted by peer reviewers to assess the quality of an article before being published and indexed in their library. However, providing an estimation of the quality of the paper can assist both authors and reviewers in providing an estimation of the number of citations that the article is expected to get per year. Authors can use such estimation to improve their articles before being submitted for publishing and reviewers to provide faster decisions. In both cases, the number of articles being queued for revision then rejected is reduced, which improves the efficiency of the process and reduces the time between the submission of an article and its acceptance.

In this study, a method is proposed to measure the quality of the article and provide an estimation for the number of citations the article gets per year. The proposed method uses artificial neural networks to predict the quality based on the title, abstract and keywords of an article. Two approaches are used to fuse the predictions of the neural networks in the model, one uses statistical functions and the other uses feed-forward neural network. In the statistical-based approach, each neural network is trained to predict the quality measure. Then, these values are used by the statistical function to calculate the overall quality measure. In the machine learning approach, a single hybrid neural network is implemented, where the title, abstract and keywords are fed to the network and a single quality measure is outputted from the network. For the fusion stage, a feed-forward neural network collects the outputs of the three networks and fuse them in a single value.

The experimental results show that the use of a single hybrid neural network to directly predict the quality of the article is more accurate than processing each part of the article, separately, then combine them into a single measure using statistical methods. Moreover, the use of one-dimensional convolutional layers in the neural network to process the title and abstract data has shown the highest performance, with only 2.52 MSE. The closest competitor to this type of neural networks is the LSTM, which has achieved MSE of 7.66. Both the convolutional and the LSTM networks have been able to outperform the simple-RNN network in all experiments, according to their ability of detecting and combining

words from different positions from the sentences. In simple-RNN network, the position of the word has a significant effect over the computation of the current position, where closer words have higher effect on these computations.

In future work, more information is collected from each article and used for quality measures. Such information may include the conclusion or the experimental results presented in the article. Although the use of more information can produce more accurate measures, this addition increases the complexity of the computations required to produce the quality measure.

REFERENCES

[1] E. A. Fox, "Introduction to digital libraries," in *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, 2016, pp. 283-284.

[2] L. Bornmann and R. Mutz, "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references," *Journal of the Association for Information Science and Technology,* vol. 66, pp. 2215-2222, 2015.

[3] A. Chakrabarti, "Growth and Development of Indian Institutional Digital Repositories in OpenDOAR: A Bird's Eye View," *World Digital Libraries,* vol. 10, pp. 75-88, 2017.

[4] D. Nicholas, A. Watkinson, H. R. Jamali, E. Herman, C. Tenopir, R. Volentine*, et al.*, "Peer review: still king in the digital age," *Learned Publishing,* vol. 28, pp. 15-21, 2015.

[5] V. M. Nguyen, N. R. Haddaway, L. F. Gutowsky, A. D. Wilson, A. J. Gallagher, M. R. Donaldson*, et al.*, "How long is too long in contemporary peer review? Perspectives from authors publishing in conservation biology journals," *PloS one,* vol. 10, p. e0132557, 2015.

[6] P. Perakakis, A. Ponsati-Obiols, I. Bernal, C. Mosquera-de-Arancibia, C. Sierra, N. Osman*, et al.*, "Open Peer Review Module (OPRM)," 2016.

[7] A.-W. Harzing and S. Alakangas, "Google Scholar, Scopus and the Web of Science: a longitudinal and cross-disciplinary comparison," *Scientometrics,* vol. 106, pp. 787-804, 2016.

[8] I. Tahamtan, A. S. Afshar, and K. Ahamdzadeh, "Factors affecting number of citations: a comprehensive review of the literature," *Scientometrics,* vol. 107, pp. 1195-1225, 2016.

[9] B. I. Hutchins, X. Yuan, J. M. Anderson, and G. M. Santangelo, "Relative Citation Ratio (RCR): A new metric that uses citation rates to measure influence at the article level," *PLoS biology,* vol. 14, p. e1002541, 2016.

[10] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science,* vol. 349, pp. 255-260, 2015.

[11] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics,* vol. 13, p. 431, 2017.

[12] S. Kalmegh, "Analysis of weka data mining algorithm reptree, simple cart and randomtree for classification of indian news," *International Journal of Innovative Science, Engineering & Technology,* vol. 2, pp. 438-446, 2015.

[13] A. M. Shahiri and W. Husain, "A review on predicting student's performance using data mining techniques," *Procedia Computer Science,* vol. 72, pp. 414-422, 2015.

[14] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data mining and analytics in the process industry: the role of machine learning," *IEEE Access,* vol. 5, pp. 20590-20616, 2017.

[15] M. Tkáč and R. Verner, "Artificial neural networks in business: Two decades of research," *Applied Soft Computing,* vol. 38, pp. 788-804, 2016.

[16] I. N. Da Silva, D. H. Spatti, R. A. Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves, "Artificial neural networks," *Cham: Springer International Publishing,* 2017.

[17] L. Mou, G. Li, L. Zhang, T. Wang, and Z. Jin, "Convolutional neural networks over tree structures for programming language processing," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[18] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine,* vol. 13, pp. 55-75, 2018.

[19] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882,* 2014.

[20] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[21] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *arXiv preprint arXiv:1412.1058,* 2014.

[22] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing & Management,* vol. 50, pp. 104-112, 2014.

[23] S. Vijayarani, M. J. Ilamathi, and M. Nithya, "Preprocessing techniques for text mining-an overview," *International Journal of Computer Science & Communication Networks,* vol. 5, pp. 7-16, 2015.

[24] T. Singh and M. Kumari, "Role of text pre-processing in twitter sentiment analysis," *Procedia Computer Science,* vol. 89, pp. 549-554, 2016.

[25] T. Verma, R. Renu, and D. Gaur, "Tokenization and filtering process in RapidMiner," *International Journal of Applied Information Systems,* vol. 7, pp. 16-18, 2014.

[26] S. Vijayarani and M. R. Janani, "Text mining: open source tokenization tools-an analysis," *Advanced Computational Intelligence: An International Journal (ACII),* vol. 3, pp. 37-47, 2016.

[27] A. J.-P. Tixier, "Notes on Deep Learning for NLP," *arXiv preprint arXiv:1808.09772,* 2018.

[28] A. Kulkarni and A. Shivananda, "Deep Learning for NLP," in *Natural Language Processing Recipes*, ed: Springer, 2019, pp. 185-227.

[29] S. K. Esser, R. Appuswamy, P. Merolla, J. V. Arthur, and D. S. Modha, "Backpropagation for energy-efficient neuromorphic computing," in *Advances in Neural Information Processing Systems*, 2015, pp. 1117-1125.

[30] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in neural information processing systems*, 2014, pp. 2924-2932.

[31] D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *International Conference on Machine Learning*, 2015, pp. 2113-2122.

[32] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in neuroscience,* vol. 10, p. 508, 2016.

[33] K. B. Nahato, K. N. Harichandran, and K. Arputharaj, "Knowledge mining from clinical datasets using rough sets and backpropagation neural network," *Computational and mathematical methods in medicine,* vol. 2015, 2015.

[34] B. Kayalibay, G. Jensen, and P. van der Smagt, "CNN-based segmentation of medical imaging data," *arXiv preprint arXiv:1701.03056,* 2017.

[35] Y. Lu, S.-C. Zhu, and Y. N. Wu, "Learning frame models using cnn filters," *arXiv preprint arXiv:1509.08379,* 2015.

[36] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *arXiv preprint arXiv:1511.05879,* 2015.

[37] J. Xu, W. Peng, T. Guanhua, X. Bo, Z. Jun, W. Fangyuan*, et al.*, "Short text clustering via convolutional neural networks," 2015.

[38] D. Chen, J. Bolton, and C. D. Manning, "A thorough examination of the cnn/daily mail reading comprehension task," *arXiv preprint arXiv:1606.02858,* 2016.

[39] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329,* 2014.

[40] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104-3112.

[41] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101,* 2016.

[42] A. Kuncoro, M. Ballesteros, L. Kong, C. Dyer, G. Neubig, and N. A. Smith, "What do recurrent neural network grammars learn about syntax?," *arXiv preprint arXiv:1611.05774,* 2016.

[43] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[44] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," 1999.

[45] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *ACM SIGMETRICS Performance Evaluation Review,* vol. 41, pp. 70-73, 2014.

[46] D. Hovy, B. Plank, and A. Søgaard, "When POS data sets don't add up: Combatting sample bias," in *LREC*, 2014, pp. 4472-4475.

[47] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 32, pp. 569-575, 2010.

[48] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," *Journal of machine learning research,* vol. 5, pp. 1089-1105, 2004.

[49] T. Fushiki, "Estimation of prediction error by using K-fold cross-validation," *Statistics and Computing,* vol. 21, pp. 137-146, 2011.

[50] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 5, pp. 216-233, 2015.

[51] C. Blake, "UCI repository of machine learning databases," *http://www. ics. uci. edu/~ mlearn/MLRepository. html,* 1998.

[52] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649-657.

[53] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[54] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2428-2437.

[55] I. Banerjee, Y. Ling, M. C. Chen, S. A. Hasan, C. P. Langlotz, N. Moradzadeh*, et al.*, "Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification," *Artificial intelligence in medicine,* 2018.

[56] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos*, et al.*, "Deep complex networks," *arXiv preprint arXiv:1705.09792,* 2017.

[57] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802-810.

[58] A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4013-4021.

[59] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: A deep neural network that beats humans," *International journal of computer vision,* vol. 122, pp. 411-425, 2017.