Republic Algeria Democratic and Popular
Ministry of Higher Education and Scientific Research
University of El Tarf
Faculty of Science and Technology
Department of Computer Science

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

MASTER THESIS IN COMPUTER SCIENCE

Presented By

Aoun Nada

SPECIALITY : Intelligent Computer Science Systems

THEME

Design and implementation of an online education platform (open classroom)

Presented on: 23 / 06 / 2024

Qualité	**Nom et Prénom**	**Grade**	**Université**
President	Mr.Djedai A	MCB	El-Tarf ,University
Supervisor	**Mr. Benmachiche A**	**MCA**	**El-Tarf ,University**
Examiner	Mr. Boutouil L	MCB	El-Tarf,University

University Year : 2023/2024

Acknowledgment

My thanks go first to God Almighty for the will, health and patience he gave me during

all these years of study.

In particular, I would like to thank ***Dr Abdel madjid Benmachiche****,*

who supervised me throughout this thesis,

for all his valuable advice.

for his active listening, his availability.

Thank you for enriching our knowledge and guiding us all this time.

You are the teacher who managed to inspire me, to give me confidence in myself and in

the future, but also who managed to make me want to know more.

Thanks for everything you've done!

of this work.

I sincerely thank:

- for having honored me to chair the jury

- for agreeing to be an examiner.

To the teachers of our university and especially those of the computer science

department.

A big thank you to all my family for their concern and concern for me, their

encouragement and followed them with patience during the course of my studies.

My thanks also go to my colleagues, for the good times we passed together. Finally I

thank all those who helped me from afar or from the meadows.

Thank you all, from the bottom of the heart

I dedicate this modest work

To the one who made great efforts for my happiness.
To the one who dreamed of seeing this day.
To the one who guided me and taught me the secrets of life "my father".
To the one who represents my world and my happiness.
To the one who represents the day and the light of my life.
To the one who always wakes up to see me the best.
To the one who opened the portals for me and gave me tenderness and courage.
To the one who mourns to see me happy to the one who warmly awaits this day "my dear mother".
To my sisters.
To my family in general.
To my friends.

To the entire 2024 class of the 2nd year in computer science.
To all those who have helped me from near or far.

List of tables

ANN	Artificial Neural Network.
RNN	Reccurent Neural Network.
LSTM	Long Short-Term Memory.
CNN	Convolutional Neural Network.
DBN	Deep Belife Network.
DNN	Deep Neural Network.
MLP	Multi-Layer Perceptron.

الملخص

أدى ظهور التعليم عبر الإنترنت إلى تغيير مشهد التعلّم والتعليم من خلال تجاوز قيود الفصول الدراسية التقليدية. وتوفر هذه الثورة مرونة لا مثيل لها، مما يتيح للمتعلمين الوصول إلى موارد تعليمية عالية الجودة في الوقت الذي يناسبهم. وبالتالي، فقد فتحت سبلًا جديدة للدراسة الذاتية والتطوير المهني والوصول إلى مناهج عالية الجودة.

تتمثل إحدى المشكلات المهمة التي تم تحديدها في الصعوبة التي يواجهها الطلاب في البحث والعثور على المعلومات ذات الصلة بشكل فعال في بيئات التعلم الإلكتروني. وتنبع هذه المشكلة في كثير من الأحيان من عدم كفاية وظائف البحث، مما قد يعيق عملية التعلم. يهدف هذا العمل إلى معالجة هذه المشكلة من خلال تخصيص تجربة التعلم باستخدام تقنيات الذكاء الاصطناعي. ويركز على تصميم وتنفيذ منصة للتعلم عبر الإنترنت. من خلال استخدام مجموعة بيانات وتقنيات التعلم العميق مثل الشبكات العصبية المتكررة (RNNs)، وتحديدًا الذاكرة طويلة المدى (LSTM)، من خلال التنبؤ بالكلمة التالية، يمكن للنظام تقديم اقتراحات بحث أكثر ملاءمة وإكمال تلقائي للبحث، وبالتالي تعزيز تجربة المستخدم وجعل عملية البحث أكثر كفاءة، وأظهر معدلات دقة ملحوظة وأخطاء منخفضة. فقد حقق معدل دقة 0.97 وخطأ بنسبة 15. 0.

كذلك، أبرز تطبيق دمج الشبكات العصبية التلافيفية (CNNs) لتوليد نتائج البحث والتوصيات إلى موارد أخرى ذات صلة. حيث يحقق النموذج معدل دقة يبلغ 0.73 وخطأ 0.87.

الكلمات الرئيسية : منصة التعلم عبر الإنترنت، التعلم العميق, الشبكات العصبية، الذاكرة القصيرة والطويلة المدى،التنبؤ؛ الشبكات العصبية التالفيفية,التصنيف.

Abstract

The emergence of online education has changed the landscape of learning and teaching by bypassing the constraints of the traditional classroom. This revolution offers unparalleled flexibility, allowing learners to access high-quality learning resources at a time that suits them. Consequently, it has opened up new avenues for self-study, professional development, and access to high-quality curricula.

One important issue that has been identified is the difficulty students have in searching and finding relevant information effectively in e-learning environments. This issue often stems from inadequate search functionality, which can hinder the learning process. This work aims to address this issue by personalizing the learning experience using AI techniques. It focuses on the design and implementation of an online learning platform. Through the use of a dataset and deep learning techniques such as recurrent neural networks (RNNs), specifically long term memory (LSTM), by predicting the next word, the system can provide more relevant search suggestions and autocomplete the search, thereby enhancing the user experience and making the search process more efficient, and has shown remarkable accuracy rates and low errors. It achieved an accuracy rate of 0.97 and an error rate of 0.15.

The application of integrating convolutional neural networks (CNNs) to generate search results and recommendations to other relevant resources was also highlighted. The model achieves an accuracy rate of 0.73 and an error of 0.87.

Keywords : Online learning platform, deep learning, neural networks, short and long term memory, prediction, convolutional neural networks, classification.

L'avènement de l'éducation en ligne a changé le paysage de l'apprentissage et de l'enseignement en contournant les contraintes de la salle de classe traditionnelle. Cette révolution offre une flexibilité inégalée, permettant aux apprenants d'accéder à des ressources d'apprentissage de haute qualité au moment qui leur convient. Par conséquent, elle a ouvert de nouvelles voies pour l'auto-apprentissage, le développement professionnel et l'accès à des programmes d'études de haute qualité.

Résumé

Un problème important qui a été identifié est la difficulté qu'ont les étudiants à rechercher et à trouver des informations pertinentes de manière efficace dans les environnements d'apprentissage en ligne. Ce problème découle souvent d'une fonctionnalité de recherche inadéquate, qui peut entraver le processus d'apprentissage. Ce travail vise à résoudre ce problème en personnalisant l'expérience d'apprentissage à l'aide de techniques d'intelligence artificielle. Il se concentre sur la conception et la mise en œuvre d'une plateforme d'apprentissage en ligne. En utilisant un ensemble de données et des techniques d'apprentissage profond telles que les réseaux neuronaux récurrents (RNN), en particulier la mémoire à long terme à court terme (LSTM), en prédisant le mot suivant, le système peut fournir des suggestions de recherche plus pertinentes et compléter automatiquement la recherche, améliorant ainsi l'expérience de l'utilisateur et rendant le processus de recherche plus efficace, et a montré des taux de précision remarquables et de faibles erreurs. Il a atteint un taux de précision de 0,97 et un taux d'erreur de 0,15.

L'application de l'intégration des réseaux neuronaux convolutionnels (CNN) pour générer des résultats de recherche et des recommandations vers d'autres ressources pertinentes a également été mise en évidence. Le modèle atteint un taux de précision de 0,73 et une erreur de 0,87.

Mots-clés : Plateforme d'apprentissage en ligne, apprentissage profond, réseaux neuronaux, mémoire à court et à long terme, prédiction, réseaux neuronaux convolutifs, classification.

GENERAL INTRODUCTION

The advent of the digital age has profoundly transformed our approach to education, marking the transition to innovative online learning methods. In this ever-changing landscape, the design and implementation of online educational platforms play a central role in meeting contemporary educational needs. The swift advancement of information technologies has led to an increasing need for adaptable learning modalities that accommodate the varied lives of students.

OpenClassroom is emerging as a response to this demand, positioning itself as an online educational platform that transcends the traditional boundaries of teaching. Focusing specifically on IT courses, the potential of e-learning is fully utilized offering interactive and enriching educational experiences.

The project stands out for its holistic approach, integrating a variety of teaching aids such as videos, interactive readings, hands-on projects and personalized mentoring sessions. The aim is to create an engaging virtual learning environment that promotes student participation and enables a comprehensive comprehension of information technology principles.More than simply transferring teaching online, this initiative aims to redefine the way students approach and assimilate computer science knowledge. The ultimate goal is to meet the diverse needs of the student community, while preparing these learners for the challenges of the ever-changing digital world.

E-learning is a type of delivery method used in distance education [1] can include content delivery in multiple formats, management of the learning experience , provides faster learning at reduced costs, and increased access to learning, E-learning will be the great equalizer in the new century. By eliminating barriers of time, distance, and socio-economic status. educational institutions must offer innovative programs.[2]With the advent of Information and Communication Technologies (ICT) in the field of training, E-Learning was born. This online learning method , sovolution has gone through three distinct phases :

- correspondence courses
- computer-assisted teaching based on a behaviorist approach
- blended learning: combining distance and classroom learning

to personalize learning paths. E-learning, offers active and interactive participation, free access to information, openness to all audiences, and time autonomy for learners. However, it also has its drawbacks, such as the need for Internet access and computer skills, as well as the physical absence of a teacher. It is available in three complementary modes:

- Synchronous mode: real-time interaction between participants, promoting immediate exchange and shared understanding.
- Asynchronous collaborative mode: exchanges at different times via forums or e-mail messages, offering flexibility and greater depth.
- Self-directed mode: autonomous learning with access to resources and self-assessment tools to adjust individual learning paths. [3]

Throughout this thesis, we will explore the various dimensions of the design and implementation of this online education platform. We will highlight the challenges encountered, the innovative solutions proposed and the expected impact on students' learning experiences.

The integration of deep learning into the educational framework, represents a paradigm shift . because can offer personalized learning experiences tailored to students' individual needs and learning styles.in research can also play a crucial role. because Deep learning algorithms can be used to analyze large scientific datasets, identify patterns and trends, and even generate new research ideas. This can speed up the process of scientific discovery.

The first chapter is provides an extensive overview of artificial intelligence methods applied in e-learning. It categorizes these methods, explores their impact on education, and discusses their potential to personalize learning experiences. The chapter then delves into a comparison of these methods, outlining their respective strengths and weaknesses. Through this analysis, it aims to shed light on the evolving landscape of e-learning and the role of AI within it. Overall, the chapter serves as a foundational exploration of the intersection between artificial intelligence and education, paving the way for further discussion and research.

The second chapter is the system Design, provides an in-depth exploration of the e-learning framework leveraging Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) algorithms. With a meticulous examination of the problem statement and solution, this chapter delves into the theoretical foundations and operational mechanisms of RNN and LSTM within the e-learning domain. Through detailed elucidation complemented by conceptual models, diagrams, and flow charts, the chapter offers a comprehensive understanding of how RNN and LSTM facilitate personalized learning experiences. By visually depicting the sequential execution of these algorithms, readers gain insights into their pivotal role in optimizing educational outcomes within the e-learning environment.

The final chapter the realization and implementation. It begins with an extensive discussion on the programming language Python, along with the libraries employed to facilitate experimentation. Various implementations are conducted using Python, specifically tailored to evaluate the performance of RNN and LSTM in enhancing educational processes. Illustrated examples showcase the effectiveness of these algorithms in tasks such as personalized learning and educational content recommendation. As the thesis concludes, a comprehensive summary is provided, encapsulating the key insights and contributions made throughout the research.

Finally, as a conclusion, we summarize our thesis by giving a general view of what we discussed thus far. We also provide a glimpse of the future work of this thesis.

Chapter 1 :

Sate Of The Art

1. INTRODUCTION

The previous section was the general introduction, where we gave a basic image of our context for this thesis. We discussed the context of our work. Then, we dug deeper into its main problem, which is that the advent of artificial intelligence (AI) has profoundly reshaped the e-learning landscape, offering unprecedented opportunities to personalize, optimize and enrich educational experiences. Thanks to sophisticated techniques such as recommender systems, machine learning, chatbots and simulations, AI is revolutionizing the way we teach and learn across online platforms. This introduction will briefly explore the different AI methods implemented in e-learning, highlighting their impact on personalizing education and improving learning efficiency.

This chapter also emphasizes the numerous techniques have been i.e neural network , recommandation system optimization algorithmes (Ant colony Swarm intelligence , and genetic algorithms). Later we conducted a comparison between the selected methods where we listed the advantages and disadvantages of each to better see the strengths and weaknesses.

2. RELATED WORK

In this section, we will be reviewing methods used by researchers in e-learning.

2.1. Machin learning

Artificial intelligence is a subset of machine learning . In general, machine learning is a technique that can offer the capacity to carry out tasks that are beyond the scope of direct programming. speaking, machine learning is the process of using data to train a model, which is then used make predictions.[4]

In classroom settings, a variety of machine learning approaches have been used to track students' attention and emotional states. Deng evaluated attention based on eye state classification using techniques such as K-Nearest Neighbor (KNN), Naïve Bayes Classifier (NBC), and Support Vector Machines, however it took into account just eyes open and closed. Nigel used Bayes Net and NBC classifiers to identify emotions like as engagement and boredom, but he was constrained by the quantity of cases and demographics he could analyze. Six emotions were recognized by Savva's system during in-person lectures; however, several needed to be manually identified. Although it lacked emotion detection, Mindoro's real-time attention monitoring using YOLOv3 provided educators with real-time feedback.These systems demonstrate how machine learning may be integrated to identify affective states and student engagement, each of which has its own drawbacks and room for development.[5]

2.1.1 Deep learning

Is a machine-learning technique used to model and replicate the ways in which the human brain gathers, processes, and understands information. Neural networks, multi-layer networks modeled after the structure of the human brain, are used in deep learning. To determine the key elements of the original data, it repeatedly extracts features and performs layer-by-layer analyses of the data. Deep learning outperforms conventional classification techniques in the solution of classification issues. Natural language processing and computer vision are the two most well-known uses of deep learning.[6]

- Supervised learning algorithm predicts the results using both new and historical data sets. In order to train the software, the system first uses inputs and outputs that are supplied. Over time, the system can then automatically create targets or outputs for new data sets.
- Unsupervised learning algorithms do not use labels or classifications for the data.To find trends in the data and draw conclusions or forecasts, the system analyzes the information.
- Semi-supervised learning algorithm integrates human-based trainingin which labelled internet resources are supplied to map out certain inputs and outputs more precisely with unlabeled data.[7]

2.1.1.1. Neural network

An artificial neuron network (ANN) is A technique was used to forecast a number of e-learning-related factors, including effectiveness, likelihood of finishing the course successfully, quality, participant happiness, etc.and function similarly to how neurons in the human brain.

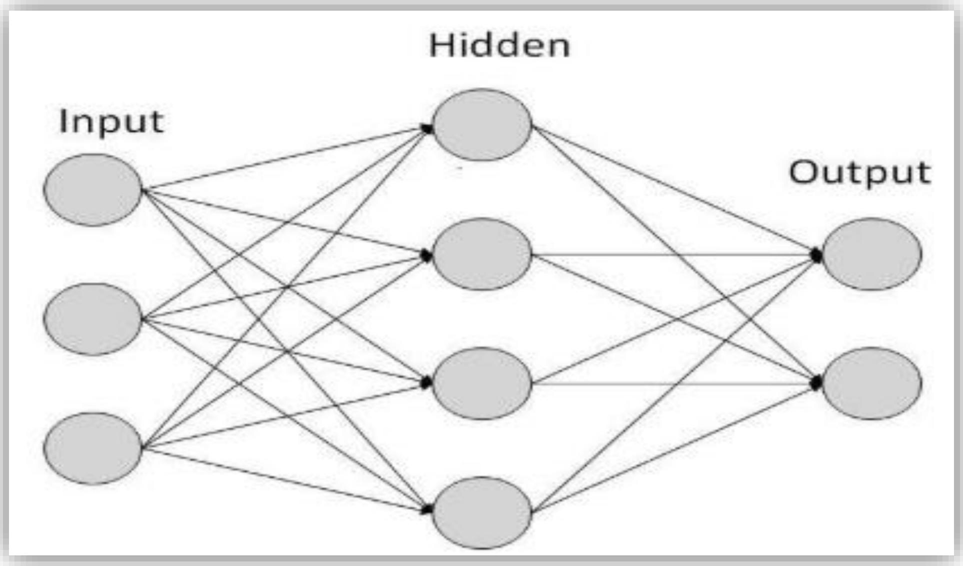


Figure 1.1: A typical artificial neural network (ANN). [8]

In e-Learning contexts, Lykourentzou, Giannoukos, Mpardis, Nikolopoulos, and Loumos studied the prediction of student grades. They realized that the growing usage of e-Learning required early prediction systems. Their study used a multilayer neural network without feedback to anticipate student marks for a 10-week online course. Based on their past academic performance, students were grouped, and the input data consisted of the outcomes of multiple choice tests. By the third week of the course, the results showed that correct forecasts were achievable with low percentages of wrong predictions, demonstrating the effectiveness of their approach. Neural networks were preferred in all forecasting phases when compared to forecasts from linear regression. This approach could help teachers provide individualized instructional support and identify.

Additionally, Baker and Richards forecasted students' educational costs using neural networks and compared the results to a prediction derived from a multivariate regression model created by the National Center for Educational Statistics. According to their findings, linear neural networks performed more accurately than statistical models.

Last but not least, Zhong, He, and Nan suggested a nonlinear estimate technique using neural networks to assess the caliber of education attained by graduating students.[9]

According to Jafari-Marandi et al., categorization tasks are essential to research, business, industry, and health care systems; as they are so widely used, even a minor change can have a significant impact. They believe that ANN is one of the most effective methods for categorization utilized in many different fields.[10]

A. Convolutional Neural Network (CNN) :

Is an amalgam of ANNs and deep learning techniques.A feedforward neural network is [11]. Its artificial neurons perform exceptionally well for processing massive amounts of images, and they may react to a portion of the surrounding units within the coverage region. Because convolutional neural networks are made up of neurons with bias constants and learnable weights, they resemble artificial neural networks. However, by introducing the weight sharing mechanism and encoding particular qualities into the network structure, the convolutional neural network improves the efficiency of the feed forward function while lowering the number of parameters. Furthermore, the Convolutional neural network's down-sampling operation can effectively expand the network's receptive area and, to some extent, provide translation invariance.[12]

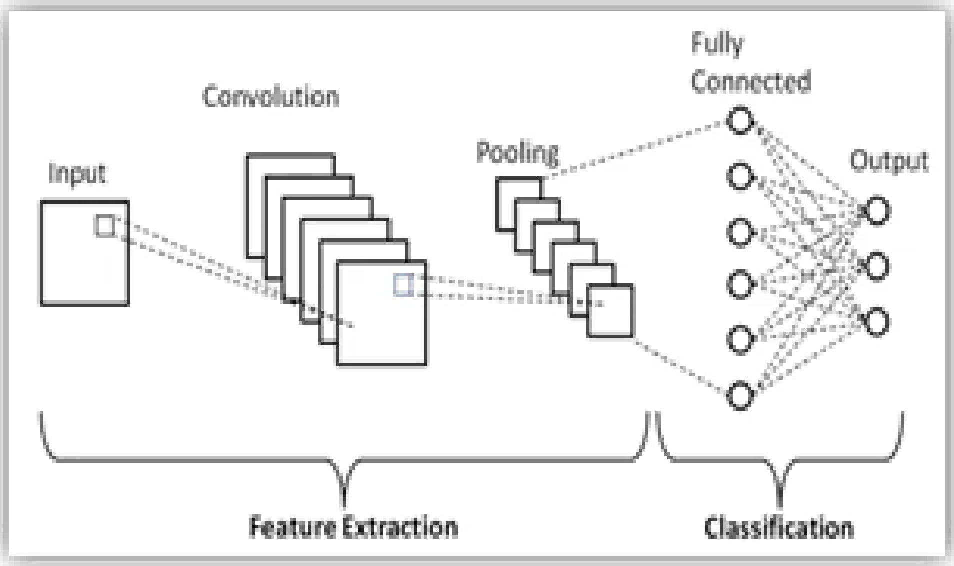


Figure 1.2 : Convolution layer[13]

Zhao et al. suggested an intelligent analytical method for behavior training based on the merging of multidimensional features. Four essential components make up the multidimensional feature fusion intelligent teaching behavior analysis mode that is produced by this method: The use of visual auditory feature-based instruction .The teaching analysis process includes the use of a behavior analysis coding system, auditory feature recognition of teaching conduct, visual feature recognition of teaching behavior, and visual data display. Then, they put forth three feasible approaches: "fusion feature-based," "visual feature-based," and "auditory feature-based." Initial analysis was done on 43 classroom Using instructional videos, the visual traits of teaching behavior are extracted in the temporal dimension to serve as a guide for tasks involving intelligent analysis of teaching behavior, including "one teacher, one pedagogy." Although this method has a solid theoretical base and is widely used in practice, it still requires improvement.[14]

Canedo used multitask cascaded convolutional neural networks (CNN) to analyze head positions captured from a camera in order to assess the attention ratings of the students. The approach's shortcomings stem from the need for substantial data and processing capacity. Moreover, the face and motion detection capabilities of Kinect sensors have been utilized by three-dimensional (3D) vision cameras to evaluate attention. Unfortunately, the study only included a row of pupils rather than the complete class due to the 3D vision cameras' technical limitations. [15]

B. Recurent Neural Network (RNN)

It is a neural net architecture that can learn sequences and handle time dependencies. It also features recurrent connections between hidden states. Recurrent connections are employed to identify links over time as well as between inputs. As a result, it works well for health issues where modeling changes in clinical data over time is common.[16](that is, strongly connected in different circumstances).[17]

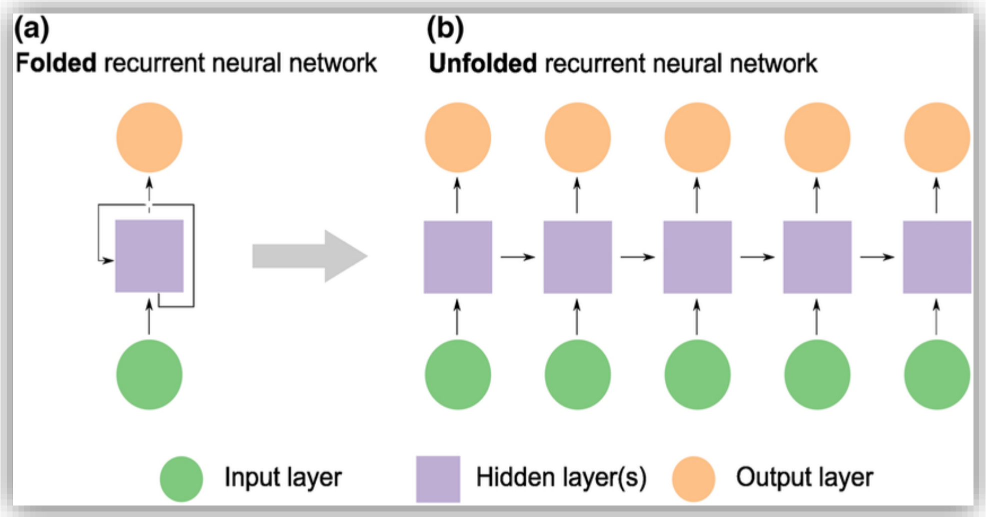


Figure 1.3 : Schematic architecture of a recurrent neural network[18]

When the data sequence lengthens, RNN decreases the memory between data. As the sequence data rises, the gradient feedback of the hidden layer should theoretically decrease layer by layer.[19]

Zhang et al. presented a content-based filtering method and a "Recurrent Neural Network (RNN)" for use in a course recommendation system. In order to determine the ranking of each course, the system first gathers information on student enrollment and course characteristics. Next, it applies a content-based filtering algorithm to identify courses that are similar to each course that students have enrolled in. This information is then sent to the network (RNN).[20]

B.1. Long Short Term Memory network (LSTM)

A variation of the RNN called the Long Short Term Memory network, or LSTM, was developed to address some of the issues that traditional RNNs encountered. One such issue is the "Vanishing gradient" problem, which is particularly sensitive to short-term[21] events and causes gradients to become extremely small or even zero when sequences are large, preventing learning. This issue is resolved by employing an LSTM-provided "Gates" mechanism to control remembered and lost data. An LSTM cell's schematic diagram can be found in (see Figure). [22]

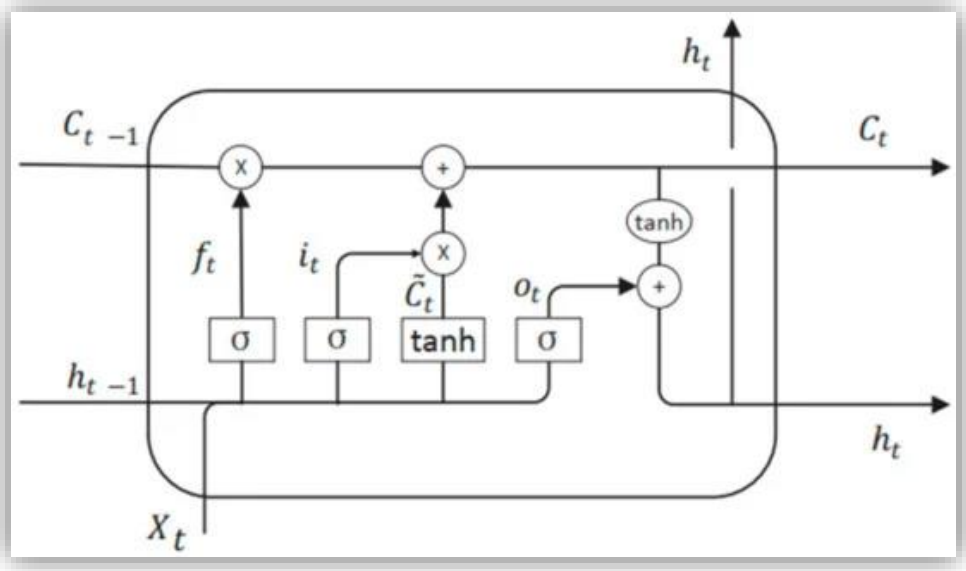


Figure 1.4 : Architecture d'une cellule LSTM[23]

The algorithm known as Long Short-Term Memory (LSTM) yields the learner style. To ascertain the learner's level, the Random Forest classifier considers variables such the learner's profile, historical data, and assessment results. The Random Forest classifier predicts the learner level, which is related to the course complexity, based on several criteria, including the students' assessment details. The combination of the learner's style and level would dictate how adaptable the course is. Under such a system, the e-Learning course would be tailored to the learner's skill level, enabling an exceptional comprehension of the material.[24]

Sharma et al. introduced LIVELINET to gauge how lively instructional videos are. Although LIVELINET uses LSTM and convolutional neural networks to forecast how lively educational films would be, it does not take into account the behavior and biology data of e-learners. Instead, it mixes audio and visual information.

De Carolis et al. provided a technique for utilizing LSTM to assess e-learners' engagement, or level of focus. The required features were extracted from the video data using the OpenFace Toolkit. With the use of LSTM, the degree of attention was predicted for the features of eye gaze, head pose, facial expressions, and facial landmarks.[25]

Aljohani et al. deployeda deep LSTM model that was used to classify student outcomes from sequential data.[26]

Qu et al. created a framework for predicting student achievement that includes an attention mechanism and uses an LSTM neural network to represent how kids learn.[27]

C. Deep belief network (DBN)

One popular type of Deep Neural Network (DNN) is the Deep Belief Network (DBN), which we use for multimodal sensing in our proposed smart classroom. Figure 6 shows the architecture of a DBN, which is made up of several stacked Restricted Boltzmann Machines.[28]A unidirectional link exists at two levels on top of layers in this paradigm. Each sub-network's hidden layers act as a visible layer for the layer above it. [29]

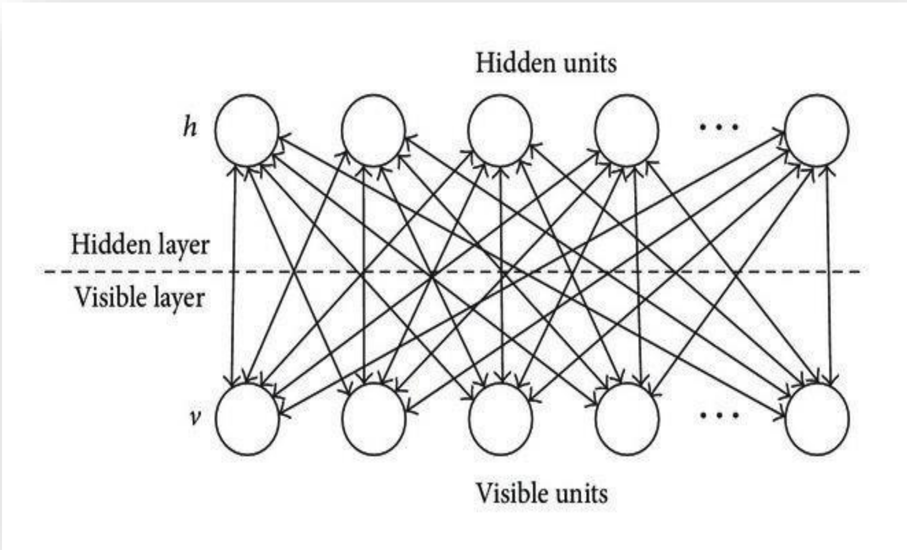


Figure 1.5 : Restricted Boltzmann Machine. [30]

Is made up of multiple hidden layers of stochastic latent variables and a layer of visible neurons.There are symmetrical links connecting the final two hidden layers, [31].

Through directed connections, the other hidden levels are connected. An RBM that uses the preceding layer's projection as an input vector pre-trains each hidden layer. Also unsupervised are DBNs. They are mostly employed for initializing a discriminant network or extracting high-level representations from incoming data. They exhibit strong performance in multiple domains, Chao & al including exchange rate prediction , Sarikaya & al used in language understanding, Lee & al used for image processing ,and Mohamed & al Sainath & al for speech processing .[32]

D. Deep neural network (DNN)

Is a deeper version of the neural network it typically has three different kinds of layers: output, hidden, and input. While the hidden layers can grow to numerous layers based on the complexity of the signal-processing method, the input and output layers are single layers. There are several nodes in each layer, and the impacts only affect layers that are next to each other.[33]

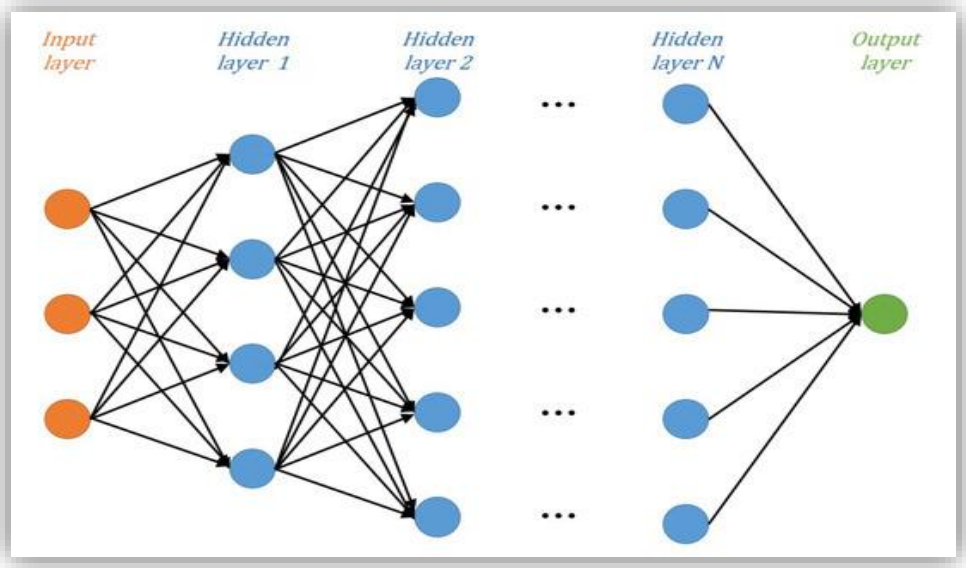


Figure 1.6 : Building the deep neural network (DNN) model.[34]

Some ideas made use of fantastic multimedia gadgets or interactive smart boards. Using voice instructions or gestures, others might operate multimedia equipment based on the task the instructor was completing. Another set of papers suggested using middleware, agent-based software, robotics, sensor networks, ubiquitous and mobile computing, multimedia computing, middleware, and AI to enhance student learning, lectures, and teaching. Few of them used deep learning algorithms to help enhance or support the classroom learning process, such as identifying hand gestures and movements to draw on canvas or suggesting to an in-class presenter that they modify their non-verbal conduct based on emotion recognition.[35]

E. Multilayer perceptron (MLP)

Is a feed-forward neural network where the input and output layers are separated by numerous (one or more) hidden layers. In this case, the perceptron does not always represent a strictly binary classifier and can use any kind of activation function. MLPs, which learn hierarchical feature representations, can be thought of as stacked layers of nonlinear transformations. Another name for MLPs is universal approximations.

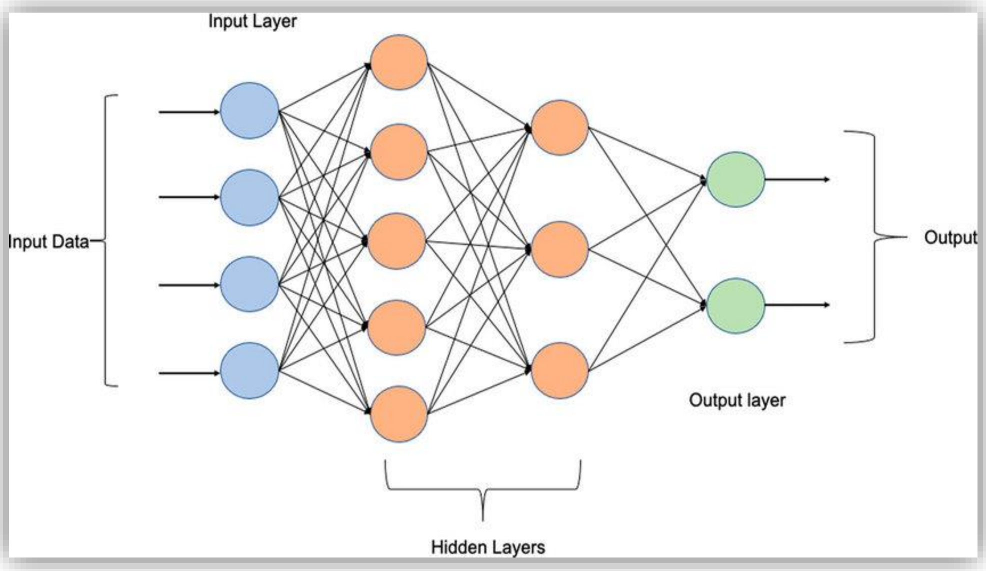


Figure 1.7 : Structure of a multilayer perceptron (MLP) algorithm,[36]

Deep learning has been drastically changing recommendation architectures lately and opening up new possibilities for recommender performance improvement. Recent developments in deep learning-based recommender systems have drawn a lot of attention since they have successfully solved problems with traditional models and produced very accurate recommendations. It is possible for deep learning to capture the non-trivial and non-linear user-item relationships. In order to find the optimal recommendation, a deep learning method called MLP-feed forward 4 layer is proposed in this study.

Al-zahrani et al use a machine learning paradigm to identify fraud in online exams.[37] Castro et al use an MLP to forecast students' performance in an online course. [38] , Iwendi et al. proposed an MLP-based tricherie detection method for multiple-choice online exams.[39]

*F. **Bayesian network***

Is a directed graph where the edges are the sporadic or significant connections between the variables and the nodes are the uncertain variables of interest. There will be a node probability table, with conditional probability values, associated with each node.

Judea Pearl proposed two processes involved in learning path generation with Bayesian Networks. A node probability table based on Bayesian Probability Theory is created in the first step. The probability shows the likelihood of the different nodes that could come after the present node and be reached from it. Candidate learning paths are those that have this probability value assigned based on the learner's level of knowledge, learning style, and learning pace. Building a Bayesian network and calculating the probability value for each learning path knowledge unit constitute the second phase in the process. [40]

3. RECOMMENDATION SYSTEMS (RSs)

Having a significant impact on the past ten years since they give users access to pertinent and customized information. Many efforts have been made in the areas of recommender system implementation, usage, and information filtering to help people identify products that are relevant to their interests, the recommendation engine gathers the preprocessed learning materials and our created patterns to calculate their similarity and predictions for the target learner. The N most similar learning resources are used to calculate the projected ratings, which gives our final results confidence, interpretability, and robustness qualities.[41]

Numerous scholarly investigations have put forth inventive methods to augment e-learning using recommender systems.

Hishehchi et al. presented a semantic recommender system that offers learners individualized learning resources by employing OWL rules and ontology. Tan et al. used user-based collaborative filtering in online learning, providing a workflow consisting of modules for managing recommendations and handling data. By combining clustering and pattern mining, Bhaskaran et al. created an intelligent recommender that adjusts recommendations based on the learning preferences of each user. Reviewing recommender paradigms, Kulkarni et al. highlighted the potential of e-learning recommendation systems.Srivastav and Kant carried out a comparative analysis of e-learning RSs based on deep learning. They have attempted to take advantage of the ways in which deep learning-based approaches can be used to overcome the major difficulties in e-learning RSs, such as cold-start and sparsity issues.[42]

MLP-based learning recommendation , Wang et al. proposed a deep learning-based framework, improving accuracy and scalability of recommendations.[43]

Zarzour et al. introduced RecDNNing, a novel approach combining deep neural networks with embedded user and item representations for improved prediction accuracy[44]. Together, these research demonstrate the variety of techniques used from deep learning techniques to collaborative and semantic filtering and emphasize the ongoing efforts to improve individualized learning outcomes. This analysis also shows how crucial feature selection is to improving suggestion quality even more.[45]

4. OPTIMIZATION ALGORITHMS

4.1 Ant Colony Optimization

Is a family of optimization metaheuristics that draws inspiration from the behavior of ants and other organisms that work together to construct a super-organism.based on the user's profile and the pathways taken by previous users of the e-learning content, [46] is used to estimate the optimal path for the learner. An expansion of the Ant Colony system is the Attribute-based Ant Colony System (AACS). It is a technique for determining which learning resources would be

best for a student based on the learning paths that prior students took the most frequently. The system creates a strong and dynamic learning object search mechanism by updating the trails pheromones from various learning modes and knowledge levels. To do this, three requirements must be met.

a) The adaptive learning portal is aware of the learner's characteristics, such as their learning style and degree of knowledge.

b) The characteristics of the student and the learning object that the instructor or content providers have annotated

c) Aligning the connections between the learning item and the learners.[47]

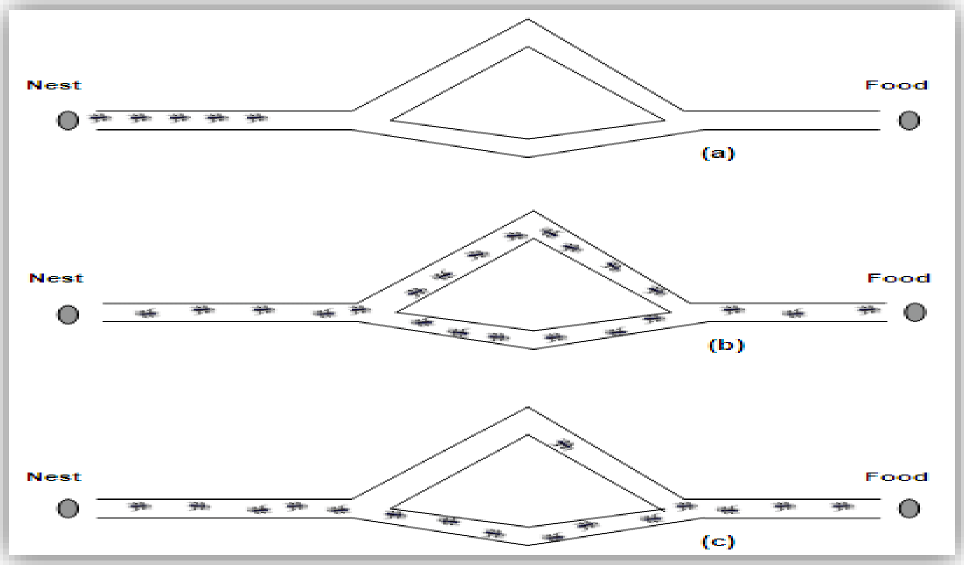


Figure 1.8 : Explanation of the Ant Colony Optimization.[48]

Chaouni et al. have proposed a three-part ACAEL (Ant Colony Adaptive E-Learning) system. The first part defines using multiple criteria evaluation (time spent consulting course units, evaluation time, number of attempts, and test score), the learner's profile is defined in the first section. The ACO algorithm is used in the second section of ACAEL to define the order of learning units. Furthermore, ACEAL's third section includes a survey component designed to evaluate instructor and student satisfaction and make improvements to the adaptive system's course content. Using graphs as a model, Bourbia et al. employed the ACO algorithm to create an adaptive e-learning platform. The graphs' nodes stood for educational materials, such as exercises or courses, and their arcs represented navigation connections. Every arc has a value that describes how important it is to teach in respect to its surrounding arc. These links are used by virtual agents, or ants, who represent students.[49]Labroche et al, have been used to guide users through a website.[50]

4.2 Swarm Intelligence

It is a type of artificial intelligence technique which is predicated on how decentralized, self-organizing systems behave together. An optimization approach for evolutionary computing is called Particle Swarm Optimization (PSO). PSO imitates social insect behavior, such as that of bees. The population of randomly initiated particles moves over the solution space, sharing the data they discover. Particles collaborate to discover a solution by dynamically adjusting their velocity based on this knowledge. Different learning objects, including basic courses, itinerary courses, required courses, and elective courses, are used to categorize domain knowledge. Learning object metadata records are updated to reflect prerequisite and learning result competences, and competency records are produced to describe learning object constraints. PSO agent settings are set once the problem has been identified in order to determine the most practical learning path. [51]

Figure **1.**9 **:** The Inspiration of the Swarm Particle Optimization.[52]

4.3 Genetic algorithms

An attempt to replicate, in a specific setting, the Darwinian model of natural evolution. Their lexicon is akin to that of natural genetics. [53]

Is a particular kind of search algorithm. It looks for the best answer to a problem by searching a solution space. The way the search is conducted is the main feature of the genetic algorithm. In order to find ever-better answers, the algorithm generates a "population" of potential solutions and allows them to "evolve" across several generations. The algorithm's bolded items are defined here.

The group of potential solutions that we take into consideration as the algorithm runs is known as the population. New members "birth" into the population and others "dying" out of the population occur during the generations of the algorithm. An individual is a singular solution within the population. An individual's fitness is a function of how "good" the solution the individual represents is. Naturally, this depends on the problem to be solved; the higher the fitness, the better the solution.. [54]

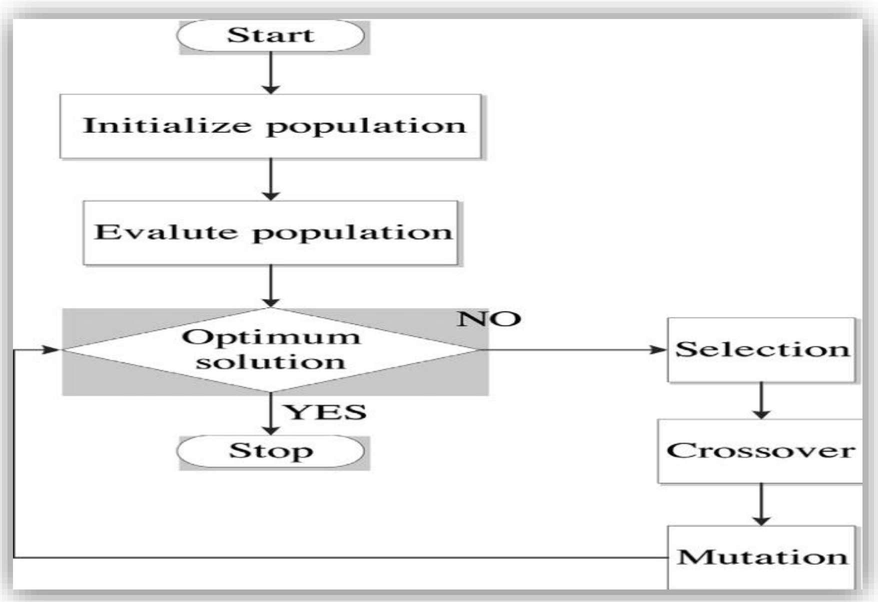


Figure 1.10: Flow Chart of the Genetic algorithm.[55]

The Application of Genetic Programming Several curricula are used in this technique to design the learning path. Every curriculum has a serial number assigned to it. There is a set beginning population size of 50. To assess the caliber of the learning path that is developed, a fitness function is created. The user's pre-test results, the curriculum's difficulty factors, and the concept relation degrees are taken into account when calculating the fitness function. To identify the next generation of learning paths, reproduction, mutation, and cross-over operations are conducted. The best-satisfied learning path is chosen after each generation of learning paths is assessed using the fitness function. [56]

5. COMPARISON BETWEEN THE METHODS

Approach	Advantages	Disadvantages
Neural network	• Adaptability to nonlinear data • Flexibility in modeling complex tasks	• Requires a large amount of data • Risk of overfitting
CNN	• Good performance for structured data • Ability to extract important features • Robustness to data variation and noise	• Sensitivity to architecture size and complexity
RNN	• Ability to process sequential data • Adaptability to variable-length sequences • Ability to capture long-term dependencies	• Sensitivity to noisy sequential data • Complexity in managing long-term context information • Complexity in managing long-term context information
DBN	• Good for extracting hierarchical features • Solid performance on massive datasets • Tolerance to noisy data	• Slow initial training • Limitations in handling high-dimensional data
MLP	• Good for simple classification problems • Can capture nonlinear relationships between features	• Can be sensitive to noisy data • Requires careful hyperparameter tuning
Recommendation	• Improvement of learner engagement	• Dependency on the quality of user data

	• Ability to recommend a variety of resources	• Requires substantial user data for personalization • May be limited by content diversity
Ant colony	• Adaptability in route optimization • Ability to handle sub-optimal solutions • Can be used in dynamic environments	• Sensitivity to environment size and complexity • Slow convergence in some cases • Requires meticulous parameter tuning • Difficulty balancing exploration and exploitation
Swarm intelligence	• Emergent collective behavior • Adaptability to dynamic and uncertain environments • Robustness to individual failure • Ability to solve complex optimization problems	• Sensitivity to population size and structure • Challenge of balancing exploration and exploitation
Genetic algithmes	• Effective exploration of search space • Adaptability to various problem types • Usable in complex research spaces	• Sensitivity to solution representation • Risk of premature convergence

Table 2 : Comparison table of the Methods .

6. *CONCLUSION*

In this chapter, we have presented the diverse methods employed for e-learning .We have seen how various researchers throughout the past two decades have put efforts We displayed those efforts and then conducted a comparison between the mentioned methods, highlighting the advantages, and disadvantages of each technique. Our aim of this chapter is to demonstrate how most of the researchers have used these soft computing techniques for elearning and how it is an important topic for the research.

We covered all that in this chapter. however, in the next chapter, we will be viewing how our system is designed, a detailed explanation of our chosen methods along with the Conceptual models that describe the system flow by creating interfaces of existing knowledge and frameworks, making it more effortless to comprehend.

Chapter 2 : System Design

1. INTRODUCTION

In the previous chapter, we examined the state of the art in the field of online learning. We explored the efforts made by researchers over the past two decades to address the challenges of implementing Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) algorithms in online learning systems.

The main object of this chapter studied algorithms. Additionally, is our approach presented conceptual models as abstract representations of how online learning tasks should be performed. These models serve as a systematic means of understanding the functioning of LSTM, and their application in online learning environments. By synthesizing existing knowledge and frameworks, our aim is to elucidate the operation of these algorithms and their practical implications for online learning systems.

2. OUR PROBLEM

In e-learning, the quest for efficiency and quality is a major concern. We focus on finding the shortest and most optimal path to quality learning. Our challenge lies in finding methods that use recurrent neural networks (RNNs) and short- and long-term memories (LSTMs) to enhance this quest. On a broader scale, we examine the use of these algorithms in optimization problems to determine their ability to find suitable solutions. We then attempt to apply them in dynamic e-learning environments, characterized by far more complex conditions, to test their effectiveness and performance. Our main question then emerges: How can RNN and LSTM algorithms render an autonomous e-learning system capable of efficiently searching and finding relevant resources, while navigating through them, while guaranteeing optimal search quality?

3. THE PROPOSED SOLUTION

In response to the question of search efficiency and quality in e-learning, our study focuses on the use of recurrent neural networks (RNNs) and short- and long-term memories (LSTMs). Our aim is to successfully adapt these algorithms to solve the challenges of finding relevant information in a dynamic e-learning environment. We aim to find efficient solutions for navigating through a massive search space, accurately identifying resources that meet learners' specific needs. Using these advanced techniques, we aim to improve the efficiency and quality of e-learning search.

4. PROBLEM DEFINITION

In the context of e-learning, we face a similar situation, albeit in a different e-learning environment. Initially, we find ourselves in a structured virtual space, defined by variables and interactions that represent the e-learning environment. Dynamic elements, such as course modules, learning activities and educational resources, are present in this space.

The objective is for learners to navigate efficiently from their starting point to their destination, while adapting to the demands of the e-learning environment. The challenges lie in finding the optimal trajectory, i.e. the most efficient path to achieve learning objectives while avoiding potential obstacles such as distractions, content gaps or conceptual difficulties.

4.1. ***Use case diagram***

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. An effective use case diagram can help your team discuss and represent scenarios in which your system or application interacts with people, organizations, or external systems the goals that your system or application helps those entities (known as actors) achieve ,and the scope of your system , UML use case diagrams are ideal for:

- articulating the objectives of system-user communication
- defining and classifying a system's functional needs
- describing a system's requirements and context
- modeling a use case's fundamental sequence of events

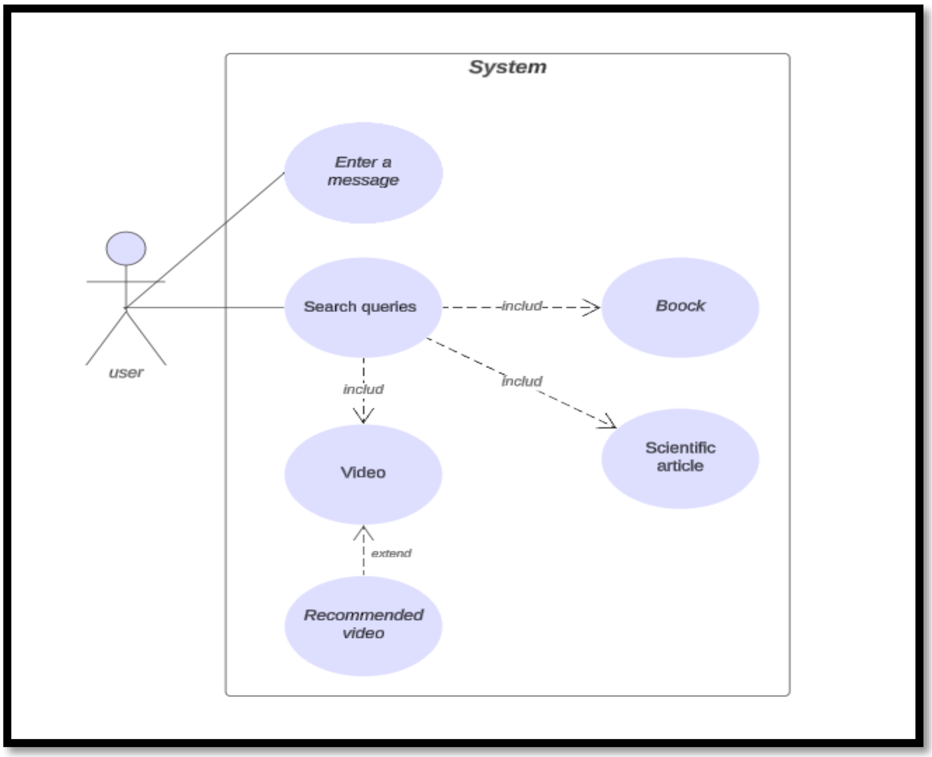


Figure 2.1 : Use case diagram.

The user can enter a message, triggering a series of search queries within the system, which can search for information in various sources such as books, scientific articles and videos. The system can also suggest recommended videos.

4.2. ***Sequence diagrams***

Sequence diagrams are a favored dynamic modeling solution in UML because they specifically concentrate on lifelines, or the processes and objects that exist simultaneously, and the messages exchanged between them to accomplish a function before the lifeline completes. The benefits of a sequence diagram are to:

- Illustrate the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- Display how objects and components interact with each other to achieve a process.
- Plan and apprehend the detailed functionality of an existing or forthcoming scenario.

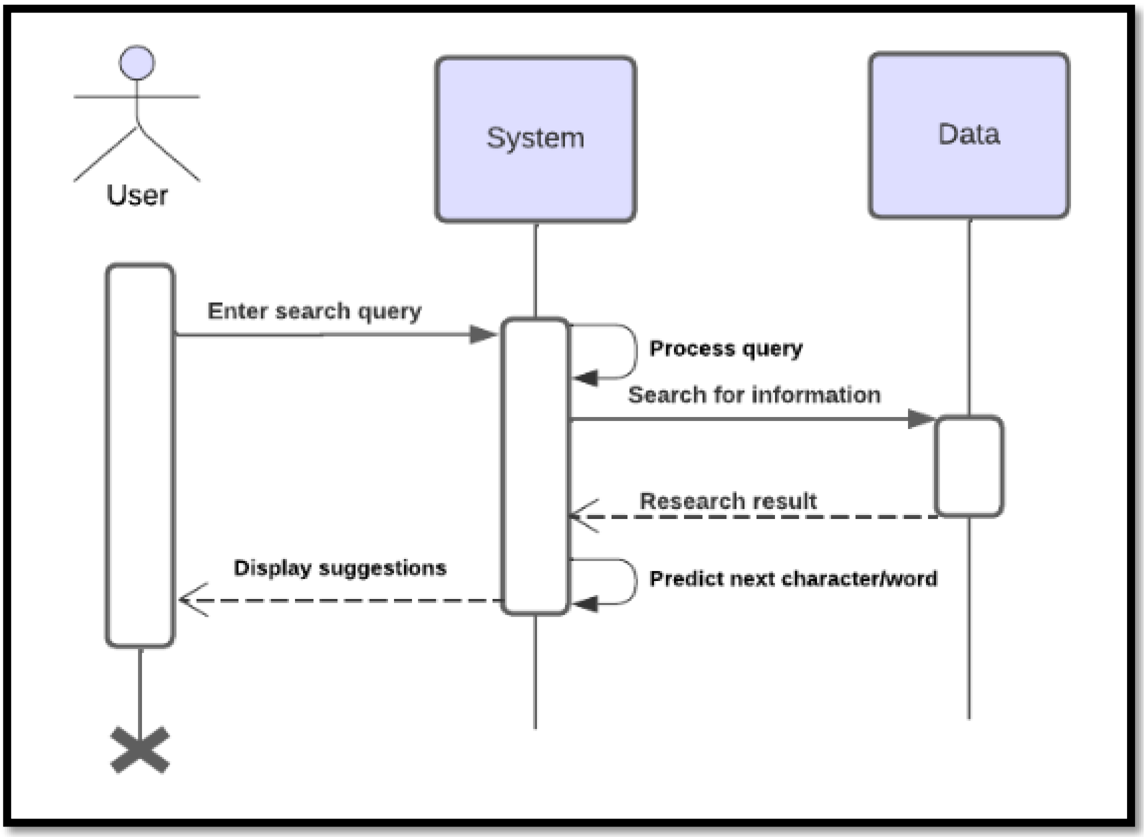


Figure 2.2 : Sequence diagram.

The user begins by entering a search query into a system. This request is then processed by the system, which searches for information in a database. Once the information has been found, the system prepares the search results and displays them to the user. If no results are found, an 'X' symbol indicates the end of the process.

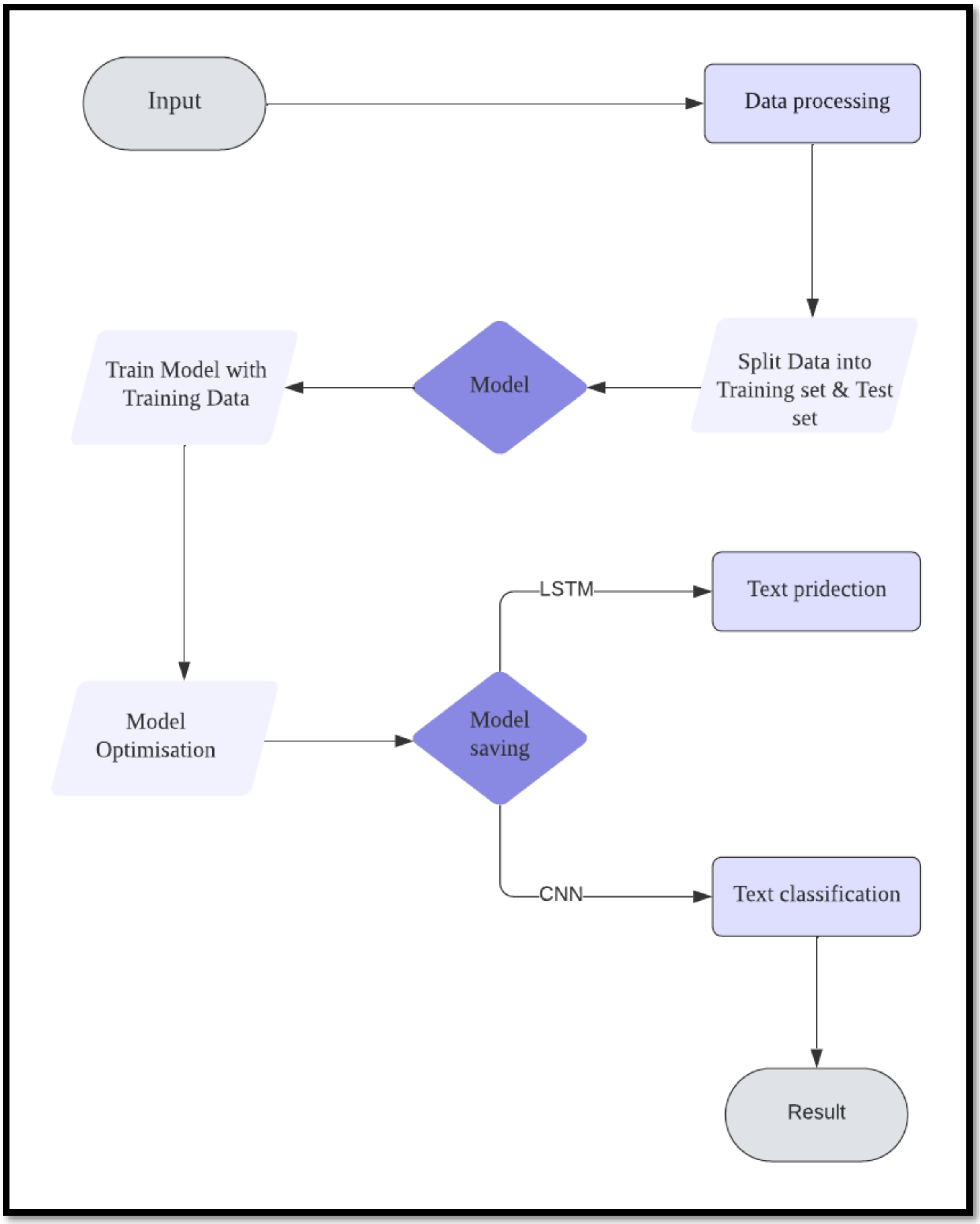


Figure 2.3 : General Architecture of Our Proposed System.

5. LONG SHORT TERM MEMORY

Information that the LSTM network determines to be pertinent to the sequence processing can be added or removed. An extra input and output are present in the LSTM cell as opposed to a simple RNN cell. We refer to this extra component as a status cell. The function of LSTM networks depends on this state cell. This cell functions similarly to a conveyor belt, allowing us to add or remove data from the network that we do not want to be there.The LSTM network is composed of the following gates:

- **Forget Gate:** It is the first gate of the cell which is responsible for the information to be thrown out or having it in the model. The sigmoid function will filter the values between 0 and 1. Previous state information and current input will be passed through the function and value nearer to 0 tends to be lost and closer to 1 means the model should retain this value.
- **Input Gate:** After passing the hidden state and current input, this information will also be passed through the tanh function to make the values between -1 and 1 to make the network send these values to the sigmoid function of this gate. Then these tanh values will be multiplied and the sigmoid function output will decide to keep the information or reject it from the tanh output.
- **Output Gate:** It is the last gate of the LSTM cell and it will decide the next hidden state which contains the information for previous inputs. In this gate the previous hidden state and current input will be passed through the sigmoid function and then cell state will be passed through the tanh function. Now, both the values will be multiplied with the sigmoid function value and it will become the next hidden state information.
- **Cell State:** At this stage, the cell state has been calculated. Cell state from forget gate will be point-wise multiplied with the forget vector. If the values will be multiplied by zero it will drop it. After this, the values from the input gate will be point-wise added which will update the new values to the cell state and the network will find this relevant.

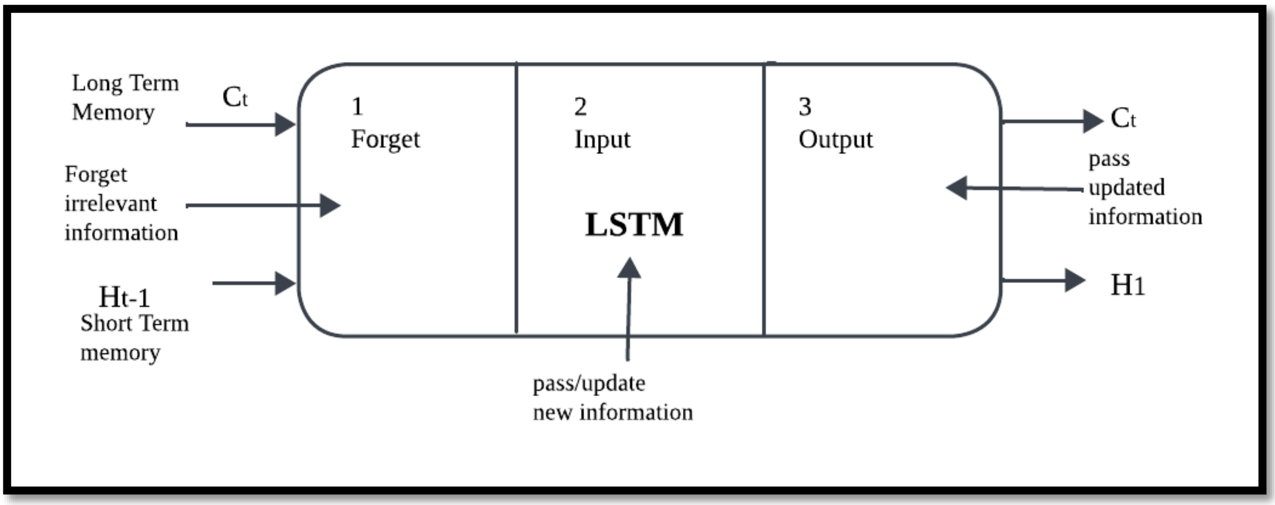


Figure 2.4 : LSTM architecture.

It's vital to remember that the LSTM network, like a simple RNN, has a hidden state where Ht is the current timestamp and H(t-1) is the past timestamp. Furthermore, as Figure 1 illustrates, the timestamps C(t-1) and C(t), respectively, , which correspond to the timestamps.

Forget gate allows us to remove items from memory, then we move on to use the update gate, which allows us to update the memory of the LSTM cell. Next, we take the previous hidden state and the current input, then transform it and bring it back to a sigmoidal activation function :

$$O_t = \sigma\,(X_t * U_f + H_{t-1} * W_t) \quad (1)$$

where: Xt represents the time stamp input; Uf represents the weight associated with the input; Ht-1 is the hidden state of the time stamp and Wf represents the weight matrix associated with the hidden state. Subsequently, a sigmoid function is applied, this causes ft to lie between 0 and 1, then it is reproduced with the current state of the previous timestamp, as shown in the (2) and (3)

$$C_{t-1} * f_t = 0 \ \ldots \ \textbf{if} \ \ f_t = 0 \ \ (\text{ Forget everything }) \quad (2)$$

$$C_{t-1} * f_t = C_{t-1} \ \ldots \ \textbf{if} \ \ f_t = 1 \ \ (\text{ Forget nothing }) \quad (3)$$

The network will forget nothing if the value of ft is 1, and everything if it is 0. Input gate functions as the input to the cell state and determines the information be written to the cell state, and is composed of two parts; 1) it passes the previous hidden state Ht-1 and the current input Xt; to a sigmoid function to determine which values to update. Then, the two inputs are passed to tanh activation to regulate the network. Finally, the output tanh Ct is multiplied by the sigmoid output in order to decide what information is important to update the cell state. In the following, (4) is presented.

$$i_t = \sigma\,(X_t * U_{fi} + H_{t-1} * W_i) \quad (4)$$

Where:

- Xt: represents the input to the current timestamp t.
- Ui: represents the input weight matrix.
- Ht-1: is the state hidden in the previous time stamp.
- Wi: is the matrix of input weights associated with the hidden state.
- In (5), if the obtained value of Nt is negative, then the information has to be subtracted from the current cell state, however, if it is positive, it is returned to the final cell state.

$$N_t = \tanh\,(X_t * U_c + H_{t-1} * W_c) \quad (5)$$

Nt, is not added directly to the state of the cell.

$$N_t = \tanh\,(X_t * U_c + H_{t-1} * W_c) \quad (6)$$

Output gate used to calculate the hidden state, this hidden state is basically a filtered version of the recently created cell state. What is done first is, to scale the new cell state to ensure that it is in the range -1 to 1, the tanh function is used to do this. Then, the output gate is used to determine which portions of the cell state will become part of the new hidden state. Finally, the values of the cell state are filtered with the vector generated by the output gate. The following is (7) of the hidden state calculation for prediction.

$$Q_t = \sigma\,(X_t * U_o + H_{t-1} * W_o) \quad (7)$$

Also, to obtain the hidden state, (7) and the updated cell state tanh are used. The (8) obtains the maximum score as an output for the prediction.

$$H_t = (O_t * \tanh\,(C_t)) \quad (8)$$

It should be noted that the hidden state is a function of the inactive or secondary memory that allows storing memories for a long-term period (Ct). To define the timestamp, we take the output of the current time or we simply apply the activation of the function softMax() Ht. Output=Softmax (Ht). here, the token with the highest score at the output is the prediction.

Dropout is a regularization technique that effectively performs NN model averaging by reducing overfitting in ANNs. It simply means that throughout a NN's training process, both visible and hidden neurons can be arbitrarily removed or omitted. In this paper, a LSTM-dropout model is proposed to improve the text prediction fit.

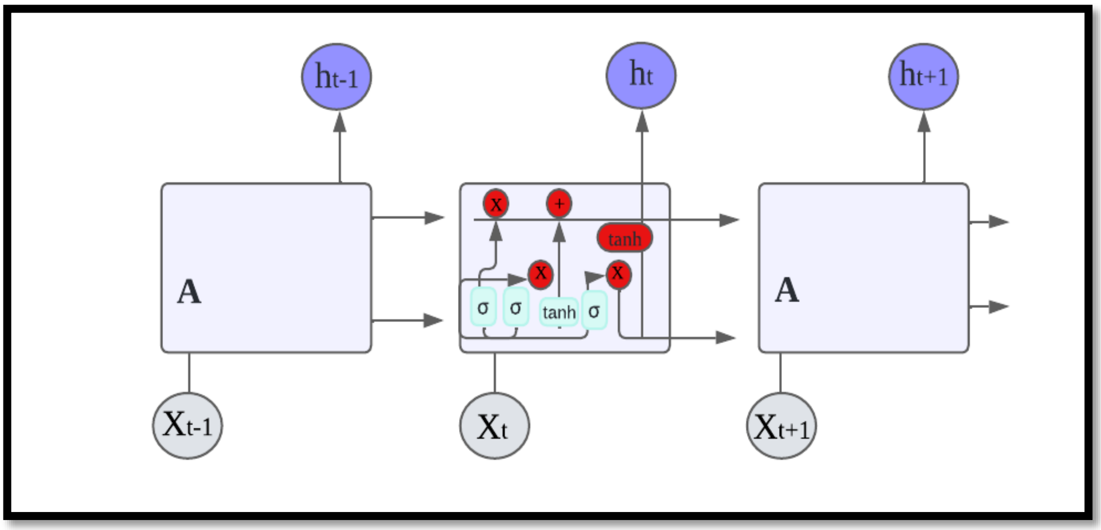


Figure 2.5 : Enhanced LSTM-dropout approach to text prediction.

5.1.LSTM PARAMETERS :

- **Text preprocessing**

  We determine a word's value based on its frequency in the same corpus. The word that has the highest importance is then selected as the context word, and we begin the processing, tokenization, which turns the data into a token sequence, as illustrated in Figure 4. The context in this model is obviously the selection of corpus terms. The method is iterative in removing terms from context and is used throughout the whole training process. The predictors are then utilized as input to forecast the following word.

- **Input layer**

  This layer has as input data the artificial neurons that incorporate a text token, which can be a complete word, a paragraph or the rot of a word .

- **Embedding layer**

  The embedding layer is defined as the first hidden layer of the network. This layer is composed of several arguments.

- **LSTM layer**

  During training, the LSTM layer performs additive interactions that serve to improve the flow in lengthy sequences. It also learns in time series and depending on sequence data. . This layer's primary characteristics are the number of hidden cells, the default output mode, it has an indicator of input and output state to the layer, it defines the input size, with respect to activations, it has an activation function to update the hidden state cell, another function to apply to input gates, among other properties. LSTM is a feedback neural network and is mainly composed of cells. In addition, the layer is able to process single data as well as sequential data, and show better performance on sequential data, such as spoken language, video and text processing .

- **Dropout layer**

  After the input layer is passed through the dropout layer to avoid overfitting in the neural network, this technique helps to select random neurons to be ignored during training, this means that neurons are temporarily removed in the next step and weight updates are not applied to the neuron in the backward step. this method helps to regularize model learning by enabling the training network to take into account all LSTM layer inputs equally rather than concentrating on only one of them.

- **Dense and activation layers**

The dense layer's job is to combine all of the LSTMs' outputs into a single value. It is closely linked to the layer before it, therefore every neuron in this layer is related to every other neuron in that layer. The activation layer's job is to change the input values that neurons receive. It accomplishes this by adding nonlinearity to neural networks, which helps the network understand the relationship between input and output values. In this model, the activation layer uses the sigmoid function. In order to estimate the likelihood of the input text, its result ranges from 0 to 1.[57]

- **Activation Functions**

It is defined as a function that is used for a neuron in a neural network to learn the underlying patterns of the dataset. Activation functions are used to introduce non- linearity in a neural network. These functions are the functions that are used in neural networks to decide to pass a value from a particular neuron or not. The most important task of an activation function is that it is used to take the summarized weighted input from a single neuron and convert it to an output value which is then fed to another deep layer if there is any. It happens during the training until the desired output is reached. In Figure is shown a graphical image of an activation function for a single neuron.

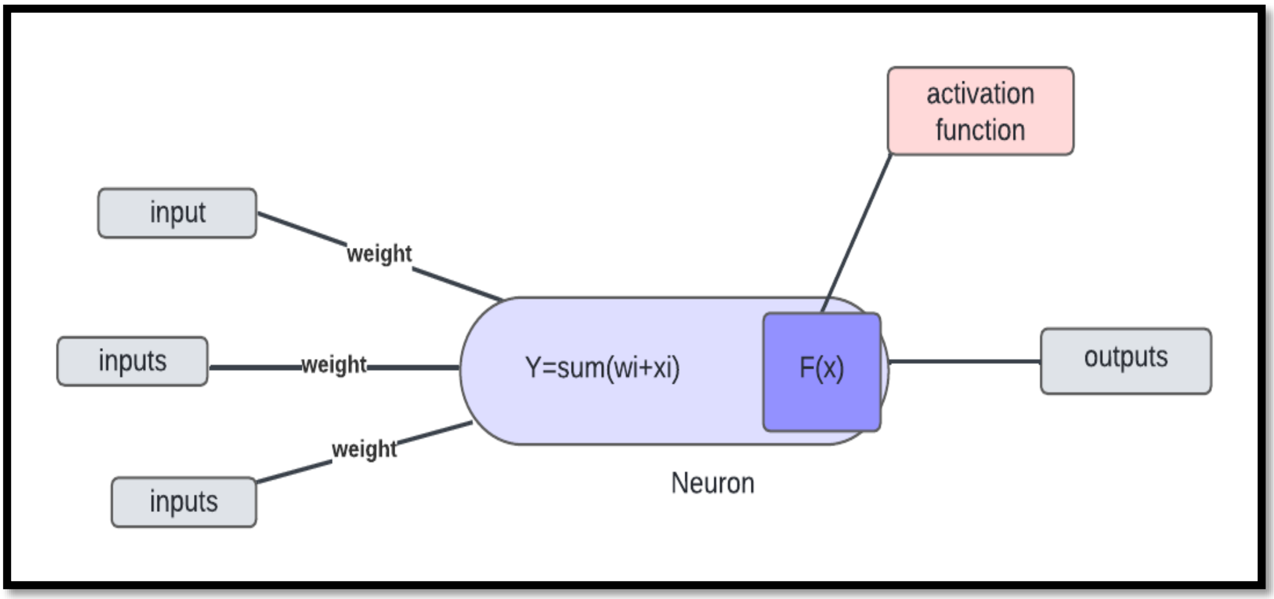


Figure 2.6 : Activation function for a single neuron.

In Neural Networks, there are three types of activation functions that are used for different models. Here are just simple definitions for these three types and after that activation functions used particularly in this thesis project will be described. The three types of activation functions are.

1. Binary Step Function.
2. Linear Activation Function.
3. Non Linear Activation Function.

- **Binary Step Function:** It is an activation function that works in a way that there must be a threshold value by which it decides that particular neuron should be activated or not. Generally, binary step functions are used for binary classification problems. In these problems, it can be decided that if a particular threshold value reaches then the output will be assigned to class one or the other class.

- **Linear Activation Function:** It is an activation function that is more flexible than the binary step function because it can deal with multiple classes. Linear activation functions are used when we need multiple outputs from a neural network e.g. when it is needed to predict multiple classes. It multiplies the weight values of each neuron and produces output from it. It takes the form of a linear function.

$$Y = w \cdot x \quad (9)$$

- **Non-Linear Activation Function:** The modern neural networks use nonlinear activation functions which are used to map complex relationships within the network to predict the output. These functions are even more flexible than linear activation functions because they introduce non-linearity in the neural network. There are multiple non-linear activation functions that can be used in a neural network. The softmax activation function will be discussed because it is used in this project.

Softmax It is an activation function that is non-linear and used to introduce non-linearity in a network. This function is used when the network is trying to solve multi-class problems. Sigmoid activation is used to solve binary classification problems.Softmax function is used to address multiple classes.[58]

5.2. LSTM MOODEL ARCHITECTURE

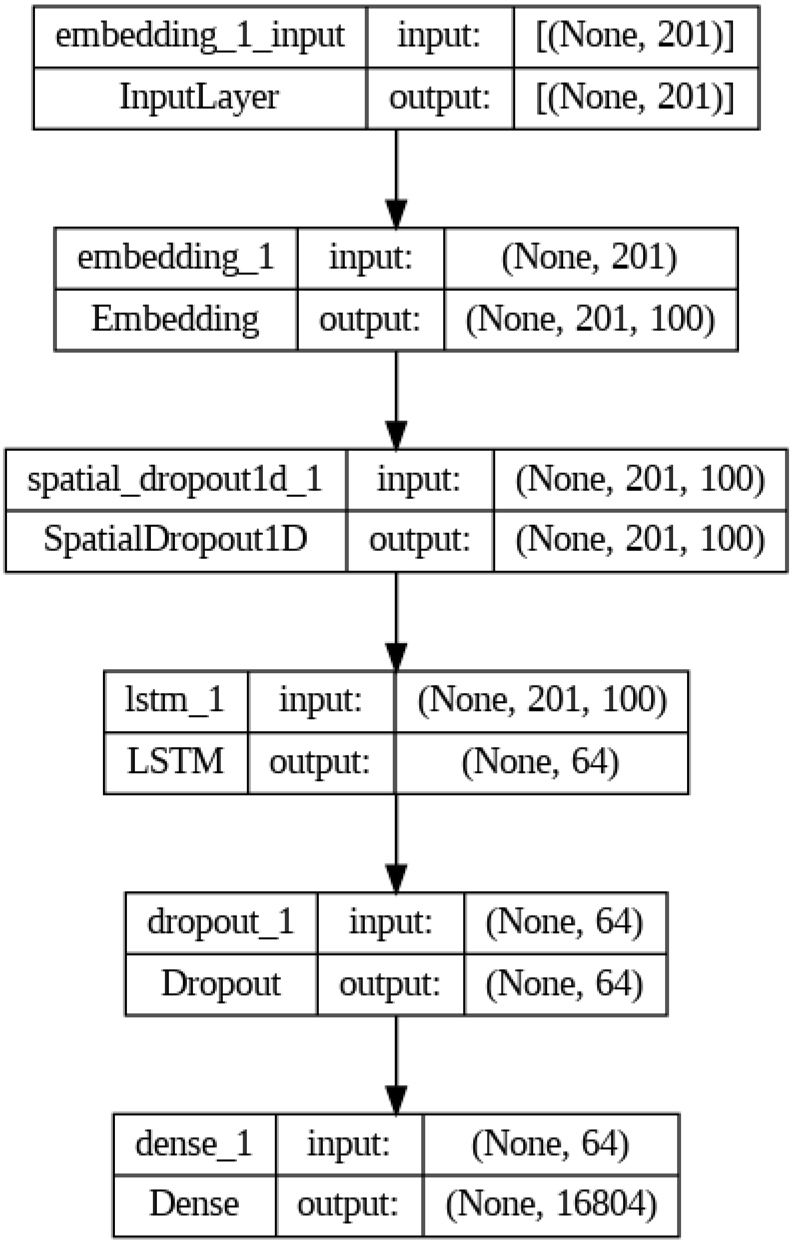


Figure 2.7 : The configuration of LSTM model

5.3. ENCODING TECHNIQUE

while text data cannot be processed by a machine. Consequently, we must transform it into a format that computers can read. Word embedding or One Hot Encoding are the two metho One Hot Encoding or Word Embedding are the two methods that can be used to accomplish this. Translating the text into vectors 1 and O constitutes one Hot Encoding. This produces a word bag that shows how frequently each word appears in the manuscript. These templates are regarded as basic ones that are highly adaptable and hold a great deal of crucial data.

Verb embedding, on the other hand, uses actual values to represent text words as a vector. Beyond 0 and 1, it can use other numbers. [59]

6. CNN MODEL

Text classification using CNN model consisting of labelled text articles. .

6.1. SENTENCE MODELLING

In each sentence, $wep \in Rz$ denotes the word embedding (a vector) for the p^{th} word in the sentence, where z is the word embedding dimension. Suppose that a sentence has n words, the sentence can now be represented as an embedding matrix $Ewe \in Rn \times z$. So we can refer to it as a word matrix where every row denotes the vectors for a particular word of the sentence. Let $wep{:}p+q$ represents the concatenation of vectors $wep, wep+1, \ldots, weq$. The convolution operation is performed on this input embedding layer. It involves a filter $k \in Rx$, that applies to a window of x words to produce a new feature. For example, a feature cp is generated using the window of words $wep{:}p+x-1$ by (1).

$$cp=(we(p{:}p+x-1).k+b) \quad (10)$$

Here, $b \in R$ and f denotes the bias and non-linear activation function respectively. The filter (kernel) k applies to all possible windows using the same weights to create the feature map (1-D vector).

6.2. PROPOSED TextConvoNet ARCHITECTURE

The proposed TextConvoNet architecture finds n-gram features between words of the different sentences and the intra-sentence n-gram feature. It is because, in the text data, having multiple sentences may have useful n-gram features. This could only be possible by using the paragraph matrix instead of the sentence matrix and applying 1-D filters. Thus, the motivation and the research question are to explore “if combining n-gram-based inter-sentence characteristics with n-gram-based intra-sentence features is beneficial or not”.The complicated way that paragraphs are put together in real-world circumstances makes it extremely challenging for any machine to identify the relevant sentiment or news category. As a result, there can be times when the model is unable to extract the inter-sentence information and produces an inappropriate result. Inspired by the aforementioned flaw, we offer a different model input structure and suggest a new CNN model that makes use of 1-D convolution and the alternative input structure .

6.3. INPUT REPRESENTATION

Each sentence is represented as two-dimensional matrix where each row represents an embedding vector for a word. Whereas in our model, the input is represented as three-dimensional matrix. In this representation, each row depicts each sentence of a paragraph, with each cell as a single word and the 3^{rd} dimension as the embeddings or the word vectors. This representation may be termed as a sentence matrix. The formal description of our input structure is mentioned below. For each sentence in a paragraph, let $Ewi \in Rz$ represents the word embedding for the i^{th} word in the sentence, where z is the dimension of the word embedding. Given that a paragraph has m sentences and n words in each sentence, the paragraph can be represented as an embedding matrix W of size (m,n,z) such that $W \in Rm \times n \times z$

The overall architecture of our proposed model, *TextConvoNet*, The presented *TextConvoNet* model uses an alternate input structure of the paragraph, using 1D convolution instead of 1D convolution and differing kernel sizes. *TextConvoNet* sends the input matrix into 4 parallel pathways of Convolution layers. The first two layers (intra-sentence layers) with 32 filters each and kernel sizes of (1 × 2 and 1 × 3), respectively, are concatenated and have the role of extricating the intra-sentence n-gram features. The other two layers (inter-sentence) with 32 filters each and kernel sizes of (2 × 1 and 2 × 2) concatenated together have the sole purpose of drawing out the inter-sentence n-gram features. These two intra-sentence and inter-sentence layers are further concatenated and fed into the fully connected layer consisting of 64 neurons and subsequently perform the relevant classification task. A detailed explanation of the architecture is given as follows.

6.4 CNN PARAMETERS :

- **Convolution layer**

This layer applies filters to the input to create feature maps and condense out the input's detected features. It is a process where we take a small matrix of numbers (called kernel or filter) and pass it over the paragraph matrix and transform it based on the values from filter. Let $Ew(m,n)$ be an input paragraph matrix of size $m \times n$ and H is a two dimensional with kernel size of $(2g + 1, 2d + 1)$, where g and d are constants. The outcome of the convolutional layer is represented by (3).

$$ri,=\sum u=-(\sum v=-ddH[u,v]F[i-u,j-v]) \quad (11)$$

Here, ri,j is the value at location (i,j) in the feature map.

- **Pooling layer**

  This layer is used to reduce the dimensionality of the data by reducing the spatial size of the representations so that the network can learn important features while retaining the essential information of the different parts of the text.

- Flattening

  This operation converts multi-dimensional data into a single dimension.Before sending the features to the complete connection layer in the context of text classification, the features extracted from the text can be flattened into a one-dimensional vector using the flattening layer.

- **Full Connection**

  Also referred to as a dense couche, this type of network is one in which every neuron is connected to every other neuron in the preceding network. A complete connection layer is frequently employed at the end of a neural network to integrate extracted features into a global text representation before moving on to the output layer for classification.

- **ReLu activation layer**

The purpose of the ReLu activation layer after each convolution layer is to normalize output. This layer also aids the model to learn something complex and complicated with a reduced possibility of vanishing gradient and cheap computation costs. The activation function for ReLu is given in (4). Here ri,j is the input to the ReLu function.

$$(ri,j)=(0,ri,j) \quad (12)$$

6.5. CLASSIFICATION

The feature maps generated by using different kernel sizes are concatenated and fed into the fully connected layer. A multilayer perceptron with connections to every activation from the layers before it is called the completely linked layer. By multiplying the weights of the matrix by an offset value, the activation of these neurons is determined. Additionally, a dropout layer is included, which helps to lessen overfitting by arbitrarily activating or deactivating (making them

0) the outgoing edges of hidden units at each update of the training phase. Ultimately, the classification layer applies a final classification based on the attributes that the earlier layers were able to extract. It's a conventional artificial neural network layer with a sigmoid or softmax activation function.

6.6. LOSS FUNCTION

For binary-class text classification task, *TextConvoNet* is trained by minimizing the binary-cross entropy(5) over a sigmoid activation function. For the task of multi-class classification, *TextConvoNet* is trained by minimizing the categorical-cross entropy (6) over a softmax activation function. The above loss functions can be formulated as :

$$BCE=-1m\sum im\sum(\sigma(y\hat{}ij)))-(1-yij)log(1-\sigma(y\hat{}ij))) \quad (13)$$

$$CCE=-1m\sum im\sum(ey\hat{}ij\sum r=1cey\hat{}ij) \quad (14)$$

Here *i* is the index of a training instance, *j* is the index of a label (class), $y\hat{}ij$ is output of the final fully connected layer, and *yij* is the ground truth (actual value) of i^{th} training sample of the j^{th} class. [60]

6.7. CNN MOODEL ARCHITECTURE

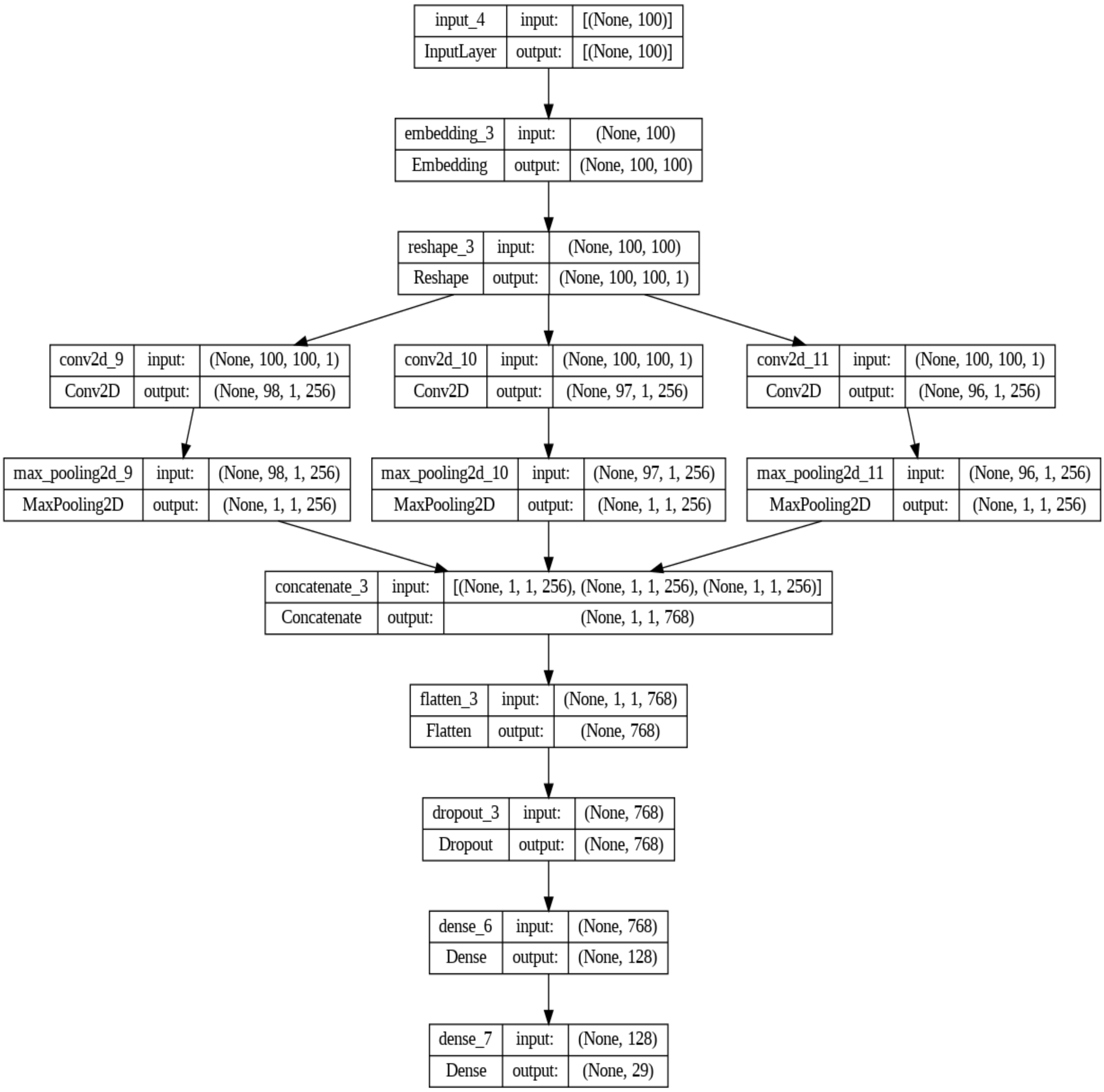


Figure 2.8 : The configuration of CNN model.

7. CONCLUSION

In this chapter, we have outlined our problem and what we consider to be an optimal solution. After the introduction, we set out the problem of search efficiency and quality in e-learning. We then proposed a solution to this problem using recurrent neural networks (RNNs) and short- and long-term memories (LSTMs). We explained in detail how these neural network architectures can be used to improve search in e-learning platforms. We presented the fundamental concepts, as well as specific methods for training and using these models in the e-learning context.general structure for our system then we went to the details of each step

As a solution, we suggested the use of LSTM to capture temporal and sequential relationships between learning data and to improve the quality of text pridection and search results in e-learning systems. We have presented the equations and internal mechanisms of these models,selected to display the results, and parameters necessary for the successful results. For further details, we built a conceptual model of the model to illustrate the knowledge and information for better comprehension.

In the next chapter, we show the tools, programming languages, and libraries we used to achieve these results. after we move on to the realization and implementation of this model in text prediction. We will present the results of our experiments and evaluate the effectiveness and quality of the searches. We will also discuss the challenges encountered when implementing these algorithms in e-learning environments, and point the way to future improvements.

Chapter 3 : Realization And Implementation

1. INTRODUCTION

In the previous section, we delved into the essence of our problem and we comprehended better , and we propose the utilization of Long Short-Term Memory (LSTM) algorithms. Afterward, we did an extensive review of our chosen method where we understood exactly how it functions and on which bases. We used the conceptual model for further explanation of the technique and how it operate. Diagrams like the use case and sequence .and highlighting the challenges surrounding efficiency and quality in e-learning search mechanisms. These algorithms offer promising solutions for optimizing search processes within e-learning platforms, leveraging their ability to discern intricate patterns in sequential data.

As we transition to a more practical perspective in this chapter, we begin by presenting the tools and equipment essential for designing and implementing LSTM-based search optimization systems within e-learning environments.

Notably, we emphasize the selection of PYTHON as our primary programming language. Python's versatility, rapid growth, and extensive support make it an ideal choice for implementing sophisticated algorithms in the e-learning domain.It is adaptable, flexible, extremely efficacious, . It is high among the fastest-growing programming language worldwide. It is adaptable, flexible, extremely efficacious, and

straightforward to employ and develop. It has an extremely active society as well. It is utilized in countless organizations due to its numerous programming paradigm asset and its

implementation of automatic memory management. Due to its extensive standard library,Python is also often referred to as a battery-included language.

Once that is clear, we move on to the implementation part of our thesis in which we display the result of our work. With the foundation set, we proceed to the implementation phase, where we demonstrate the outcomes aim to showcase the efficacy of integrating LSTM algorithms in enhancing search efficiency and quality within e-learning platforms in text pridection. Through detailed execution of our program and illustration of different phases.

2. TOOLS

2.1. Physical environment

In order to realize our system, we used this hardware :

• A laptop PC: lenovo Intel(R) Core(TM) CPU @ 1.80GHz 2.30 GHz

• RAM size 16 GB

• Operating system: 64-bit Windows 11

2.1 Software Environment:

Python is a high-level, It's an interpreted, object-oriented programming language characterized by dynamic semantics. With its built-in data structures and dynamic typing and binding, Python is highly attractive for tasks like scripting, linking, and rapid application development.

It was created by Guido van Rossum, and released in 1991. It is used for:

- Creation of web applications on a server.
- Integration with other software for workflow creation.
- Connection to database systems and file manipulation.
- Handling large amounts of data and performing complex mathematical calculations.Python can be used for rapid prototyping, or for production-ready software development.[61]

Python Syntax compared to other programming languages:

- Python emphasizes readability, resembling English and incorporating mathematical influences.
- Python uses new lines to complete commands, in contrast to other languages that often use semicolons or parentheses.
- Python's scope definition relies on indentation with whitespace, unlike languages using curly brackets for the same purpose.[62]

2.2 LIBRARIES USED

Python's standard library is exceptionally comprehensive, boasting an extensive range of functionalities, as demonstrated by the extensive table of contents below. This library encompasses built-in modules, coded in C, granting Python programmers access to system functionalities, such as file I/O, that would otherwise be out of reach. Additionally, it includes modules written in Python, which offer standardized solutions to common programming challenges. Certain modules are explicitly crafted to promote and augment the portability of Python programs.[63]

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces .[64]

Provides various tools and algorithms for natural language processing tasks, such as tokenization, stemming, and parsing, which are essential for information retrieval applications like document indexing and query processing. .[65]

TextBlob is a Python library designed for handling textual data. It offers a user-friendly API for performing various natural language processing (NLP) tasks, including part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, and others. [66]

NumPy is the essential Python package for scientific computing. A multidimensional array object, different derived objects , and a variety of routines for quick array operations—like sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and much more. .[67]

TensorFlow is an open-source machine learning library developed by Google. It is utilized for constructing and training deep learning models, as it simplifies the creation of computational graphs and enables efficient execution on various hardware platforms.[68] can also be utilized for tasks related to information retrieval, such as natural language understanding and document classification.

Keras is a high-level deep learning API, was created by Google to simplify the implementation of neural networks. Written in Python, it streamlines the process of building neural networks and offers support for multiple backend neural network computations.[69]

Keras can be used for tasks such as text classification and semantic analysis, which are relevant to information retrieval systems.

Matplotlib is a robust Python library that enables the creation of static, animated, and interactive visualizations with ease. It simplifies straightforward tasks while also providing the capability to tackle more complex visualization challenges.[70]

Difflib, a module offering sequence comparison functions, facilitates string comparisons and provides supplementary insights. Within Difflib, the SequenceMatcher class is pivotal. This class enables comparison between strings and computes a similarity score between two strings. It identifies the longest matching sequence between the strings while disregarding spaces and blank lines.[71] useful for tasks like text similarity and matching.

Elasticsearch is a search engine optimized for performance. Its distributed nature enables horizontal scaling across multiple machines, making it an ideal solution for handling large volumes of data. Typically utilized in applications requiring features such as full-text search, real-time search, and analytics, Elasticsearch offers robust capabilities in these domains.[72] A distributed search and analytics engine known for its fast search capabilities, used for indexing and searching large volumes of data.

Beautiful Soup is a Python library, simplifies the process of extracting information from web pages through web scraping. It utilizes HTML or XML parsers and offers Pythonic syntax for efficiently navigating, searching, and modifying the parsed tree structure. With Beautiful Soup, tasks like isolating titles and links from web pages become straightforward. It can extract text from HTML tags and manipulate the HTML within the document being processed.[73]

PyCharm is one of the heaviest IDEs I ever worked with, it's slow, it requires too much ram and most of all I hate its indexing time. That said, I love it.

"HyperText Markup Language (hTML) is a term that can be translated as 'hypertext markup language'. It is used to design and represent web page content and structure. .[74]

CSS stands for Cascading Style Sheets. The computer language CSS is used on the Internet to give form to HTML or XML files. As such, style sheets, also known as CSS files, contain code to manage the design of an HTML page. .[75]

The javaScript programming language is mainly used in interactive web pages and plays an essential role in web applications. In addition to HTML and CSS, JavaScript occupies a central place in the practices of web developers. .[76]

Offers a flexible solution for creating different diagrams, perfect for both startups and large corporations.[77]

2.3 PREPARING DATA

The dataset contains two files ''`train.csv,test.csv`'' , in each containing columns such as "abstract", "date", "subject" and "title". These files appear to contain information on scientific articles in various fields such as mesoscopic and nanometric physics, materials science, with details such as abstract, publication date, subject and title for each article.we selected 30000 sample.

	Unnamed: 0	abstract	date	subject	title
0	0	We study the two-particle wave function of p...	2007-03-31 00:00:00+00:00	mesoscale and nanoscale physics	Bosonic characters of atomic Cooper pairs acro...
1	1	A general formulation was developed to repre...	2007-03-31 00:00:00+00:00	materials science	Numerical solution of shock and ramp compressi...
2	2	We present recent advances in understanding ...	2007-04-02 00:00:00+00:00	strongly correlated electrons	Spectroscopic Properties of Polarons in Strong...
3	3	We describe a peculiar fine structure acquir...	2007-04-02 00:00:00+00:00	mesoscale and nanoscale physics	Filling-Factor-Dependent Magnetophonon Resonan...
4	4	We investigate the effect of tuning the phon...	2007-03-31 00:00:00+00:00	strongly correlated electrons	Tuning correlation effects with electron-phono...
...	...	...	...	...	...
176455	176455	We present a concise derivation of Landauer'...	2005-12-13 00:00:00+00:00	quantum physics	Deriving Landauer's erasure principle from sta...
176456	176456	We present a complete analysis of multiparti...	2005-12-16 00:00:00+00:00	quantum physics	Multipartite entanglement in three-mode Gaussi...
176457	176457	It is well known that, beginning in 2000, th...	2005-12-16 00:00:00+00:00	quantum physics	Present status of controversies regarding the ...
176458	176458	We introduce a new measure called reduced en...	2005-12-17 00:00:00+00:00	quantum physics	Exploring Quantum Phase Transitions with a Nov...
176459	176459	We propose a tunable on-chip micromaser usin...	2005-12-18 00:00:00+00:00	quantum physics	Persistent single-photon production by tunable...

176460 rows × 5 columns

Figure 3.1 : Data content.

The clean_text function cleans up the text by removing stopwords... Next, it converts the text to lower case, segments the attached words and finally joins the cleaned words into a single clean text string.

```
import numpy as np
import re
import wordninja
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

Figure 3.2 : Libraries used in data processing.

The libraries imported into this Python code each play a specific role in the pre-processing of textual data. Pandas is used to manipulate and analyze tabular data, while NumPy provides efficient data structures for numerical calculations. The re module is useful for searching and manipulating complex text patterns using regular expressions. Wordninja allows strings to be split into words, facilitating text clean-up. Finally, NLTK offers a range of tools for natural language pre-processing, including tokenization, empty word removal and lemmatization, essential for efficient analysis of textual data. Combining these libraries provides a complete set of tools for cleaning, pre-processing and analyzing text data.

first transforms texts into token sequences using a tokenizer, then adjusts the tokenizer on training and test texts to create a word index. It then converts the texts into sequences of tokens and padding them so that they all have the same length, facilitating their use in neural networks. The sequences are divided into features and labels, the latter being converted into class vectors (one-hot encoding). Finally, the sequences are again completed to ensure that they all have the same maximum length, guaranteeing data compatibility with the deep learning model.

3. IMPLEMENTATION

3.1. WEB SITE

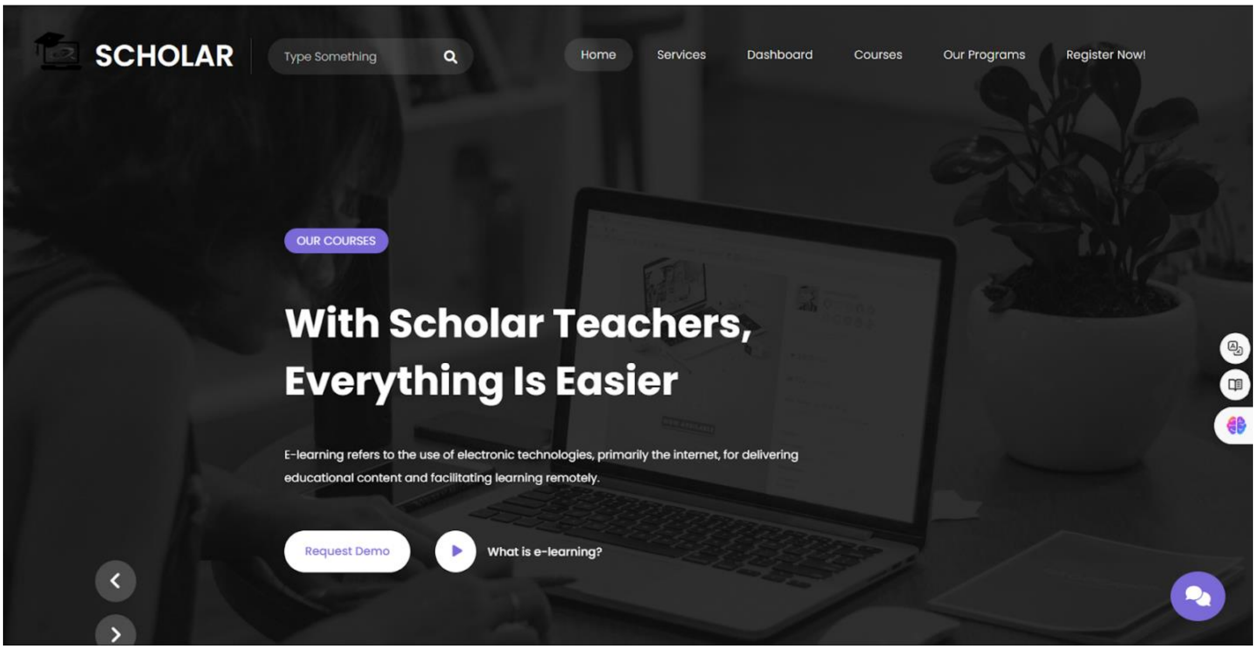


Figure 3.2 : Web site interface.

This online education site in the field of computing offers a space dedicated to the search for services and courses, giving visitors the opportunity to find the right ones for their needs and interests.

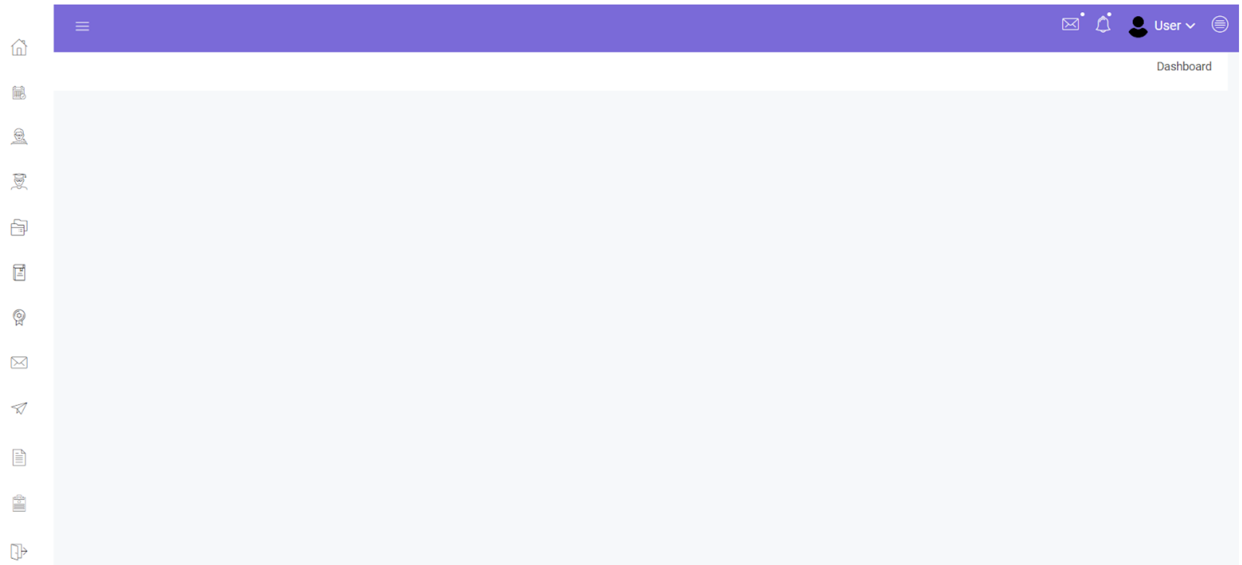


Figure 3.3 : Administration dashboard.

The interface appears to be an administration dashboard for managing teachers, students, courses and other education-related elements such as an events calendar and a library .

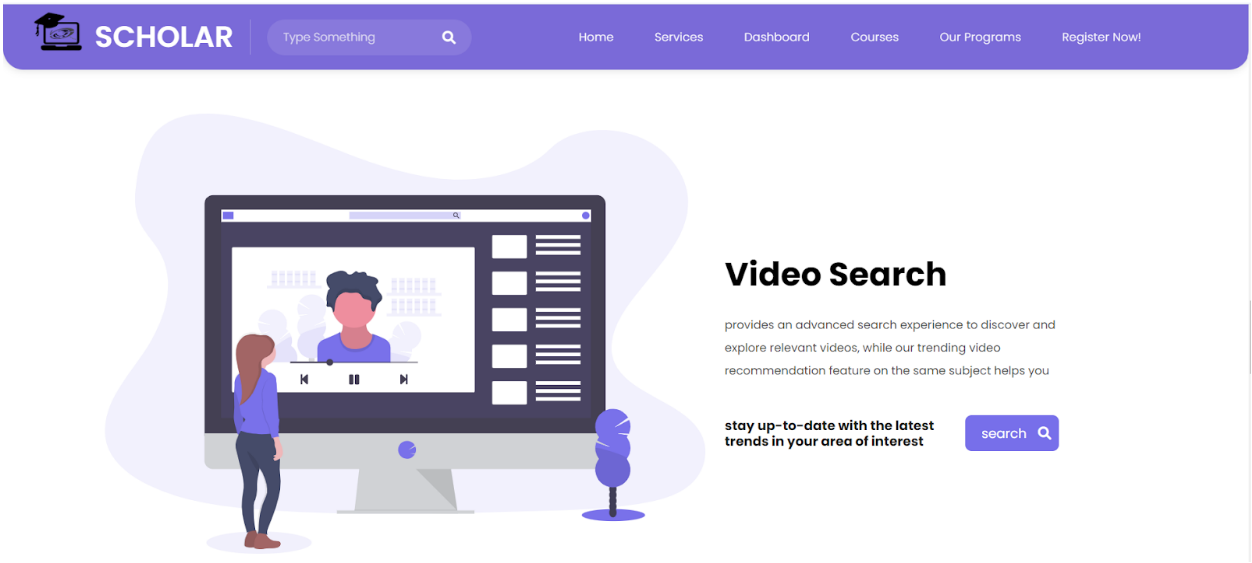


Figure 3.4 : Video search service.

This interface allowing users to search for videos on YouTube according to their queries (questions) .displays search results with their title and channel name, as well as recommended videos based on trends for the words entered.

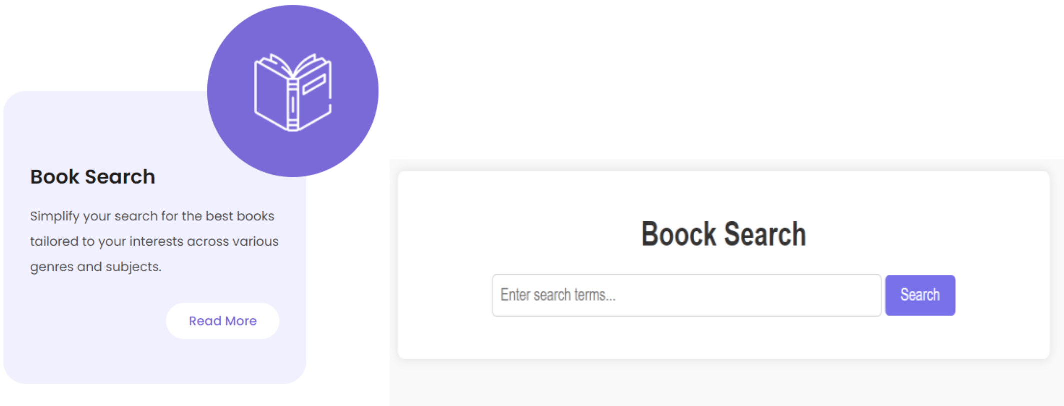


Figure 3.5 : Book search service.

The interface aims to create a fluid and intuitive user experience for searching for books online at Google Books. Users can quickly find relevant books by entering search terms. The aim is to make the book search process efficient and enjoyable by offering clear, detailed results.To achieve this, books are selected according to a number of criteria. Firstly, only books published between 2019 and 2024 are included in the results. Secondly, the relevance of search terms is paramount: books that best match the search terms entered by the user are given priority, whether in the title, description, metadata or even the content of the book. In addition, the popularity of books can play a role in their selection, measured by factors such as sales figures, positive reviews or the number of additions to reading lists. Content quality is also an important criterion. Last but not least, the topicality of books is taken into account: those that address current topics or provide up-to-date information may be favored.

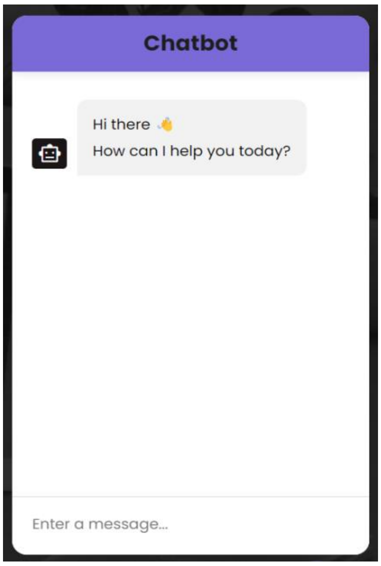


Figure 3.6 : Intelligent chat bot.

This interface implements a chatbot that uses the OpenAI API to generate intelligent replies in response to user messages via a language model (GPT-3.5 in this case). implements an Express server in Node.js to handle requests

When the server receives a request with a user message, it communicates with the OpenAI API to generate a response and sends it back to the user . and handles potential errors during communication by providing a backup response if the request fails.

3.2.EXPERIMENTS AND RESULTS

In order to present the results obtained for the model, several models were created by modifying the hyperparameters to optimize the model in the best possible way. In the following paragraphs, we present the results of accuracy and error, compared with the model architecture and the number of epochs (an epoch is the number of times the algorithm traverses the dataset).

- Accuracy: measure of how many predictions model got correct (number of correct predictions/ total number of predictions)
- Loss: measure of the error of model's predictions (usually measured as a difference between predicted and actual values)
- Validation Accuracy: the accuracy of model's predictions on a validation dataset (portion of the dataset used to assess model's performance on untested data that wasn't utilized during training)
- Validation Loss: the loss of model's predictions on the validation dataset
  Epoch:the number of epochs (complete passes through the dataset) that the model was trained for.

3.3. RNN MODEL CONFIGURATION AND NUMBER OFPARAMETER

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 203, 100)          1840500

 spatial_dropout1d_1 (Spati  (None, 203, 100)          0
 alDropout1D)

 lstm_1 (LSTM)               (None, 64)                42240

 dropout_1 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 18405)             1196325

=================================================================
Total params: 3079065 (11.75 MB)
Trainable params: 3079065 (11.75 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 3.7 : The architecture of model 1.

The model is composed of five layers first is an embedding layer that converts input words into digital vectors. The digital vectors are then used as input for the next layer lstm is a type of RNN designed to process temporal data sequences. The LSTM layer learns to capture long-term dependencies between words in a sequence. for the next layer is dropout is a regularization technique used to prevent the model from overlearning. The dropout layer randomly removes certain neurons from the network during training. Finally dense is a fully connected layer. The dense layer maps the outputs of the LSTM layer to the final output of the model. The model has a total of 4,124,904 parameters.

- Result of Model 1:

```
Epoch 18/35
313/313 - 159s - loss: 0.9855 - accuracy: 0.8724 - val_loss: 1.6532 - val_accuracy: 0.8767 - 159s/epoch - 507ms/step
Epoch 19/35
313/313 - 160s - loss: 0.8588 - accuracy: 0.8861 - val_loss: 1.5506 - val_accuracy: 0.8825 - 160s/epoch - 512ms/step
Epoch 20/35
313/313 - 160s - loss: 0.7501 - accuracy: 0.9014 - val_loss: 1.4651 - val_accuracy: 0.8875 - 160s/epoch - 511ms/step
Epoch 21/35
313/313 - 158s - loss: 0.6628 - accuracy: 0.9114 - val_loss: 1.3985 - val_accuracy: 0.8933 - 158s/epoch - 506ms/step
Epoch 22/35
313/313 - 158s - loss: 0.5880 - accuracy: 0.9189 - val_loss: 1.3416 - val_accuracy: 0.8958 - 158s/epoch - 505ms/step
Epoch 23/35
313/313 - 159s - loss: 0.5258 - accuracy: 0.9271 - val_loss: 1.2884 - val_accuracy: 0.8958 - 159s/epoch - 509ms/step
Epoch 24/35
313/313 - 157s - loss: 0.4702 - accuracy: 0.9340 - val_loss: 1.2368 - val_accuracy: 0.9033 - 157s/epoch - 500ms/step
Epoch 25/35
313/313 - 158s - loss: 0.4202 - accuracy: 0.9386 - val_loss: 1.1982 - val_accuracy: 0.9125 - 158s/epoch - 505ms/step
Epoch 26/35
313/313 - 157s - loss: 0.3788 - accuracy: 0.9440 - val_loss: 1.1626 - val_accuracy: 0.9217 - 157s/epoch - 500ms/step
Epoch 27/35
313/313 - 158s - loss: 0.3408 - accuracy: 0.9478 - val_loss: 1.1279 - val_accuracy: 0.9233 - 158s/epoch - 504ms/step
Epoch 28/35
313/313 - 156s - loss: 0.3048 - accuracy: 0.9542 - val_loss: 1.1011 - val_accuracy: 0.9292 - 156s/epoch - 497ms/step
Epoch 29/35
313/313 - 156s - loss: 0.2747 - accuracy: 0.9577 - val_loss: 1.0731 - val_accuracy: 0.9325 - 156s/epoch - 500ms/step
Epoch 30/35
313/313 - 156s - loss: 0.2505 - accuracy: 0.9610 - val_loss: 1.0501 - val_accuracy: 0.9317 - 156s/epoch - 499ms/step
Epoch 31/35
313/313 - 155s - loss: 0.2256 - accuracy: 0.9656 - val_loss: 1.0271 - val_accuracy: 0.9342 - 155s/epoch - 494ms/step
Epoch 32/35
313/313 - 156s - loss: 0.2012 - accuracy: 0.9691 - val_loss: 1.0089 - val_accuracy: 0.9350 - 156s/epoch - 498ms/step
Epoch 33/35
313/313 - 158s - loss: 0.1826 - accuracy: 0.9727 - val_loss: 0.9922 - val_accuracy: 0.9358 - 158s/epoch - 504ms/step
Epoch 34/35
313/313 - 156s - loss: 0.1683 - accuracy: 0.9739 - val_loss: 0.9792 - val_accuracy: 0.9417 - 156s/epoch - 497ms/step
Epoch 35/35
313/313 - 156s - loss: 0.1517 - accuracy: 0.9776 - val_loss: 0.9636 - val_accuracy: 0.9442 - 156s/epoch - 499ms/step
```

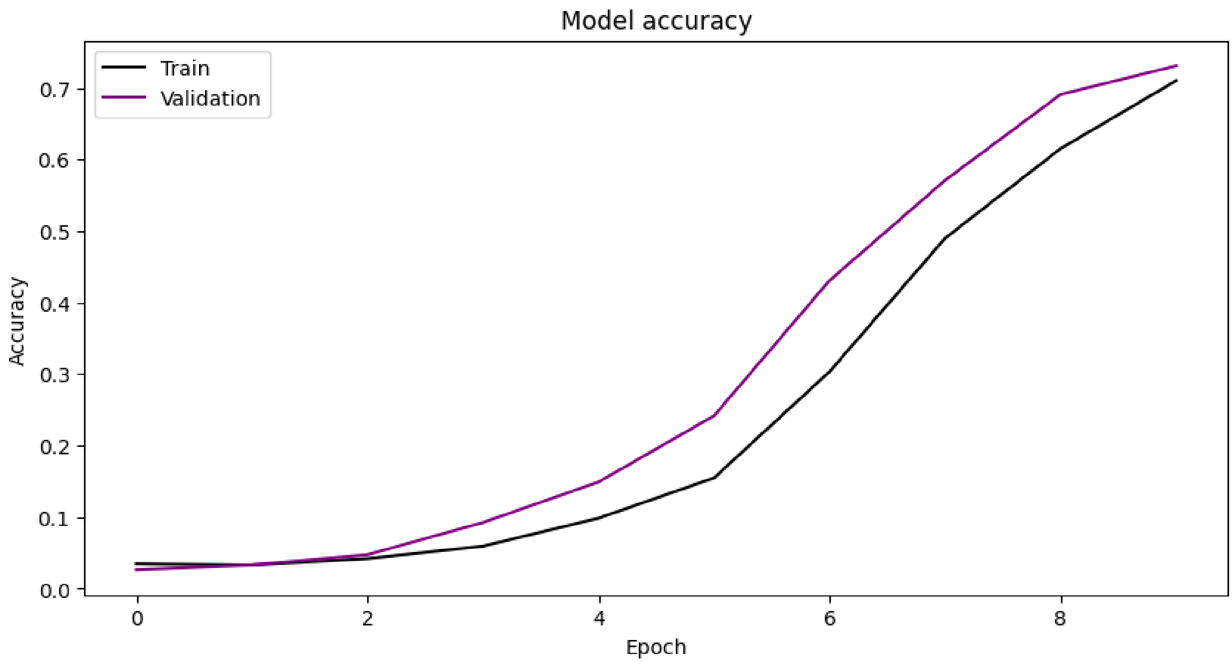
Figure 3.8 : The resulte of model 1.

3.4. COMPARISON BETWEEN DIFFERENT RESULTS

- Testing with number of epochs = 10

Epochs number	Training		Test	
	loss	accuracy	vall_loss	vall_accuracy
10	2.12	0.71	2.64	0.73

Table 2 : Result of training and testing with epoche=10.



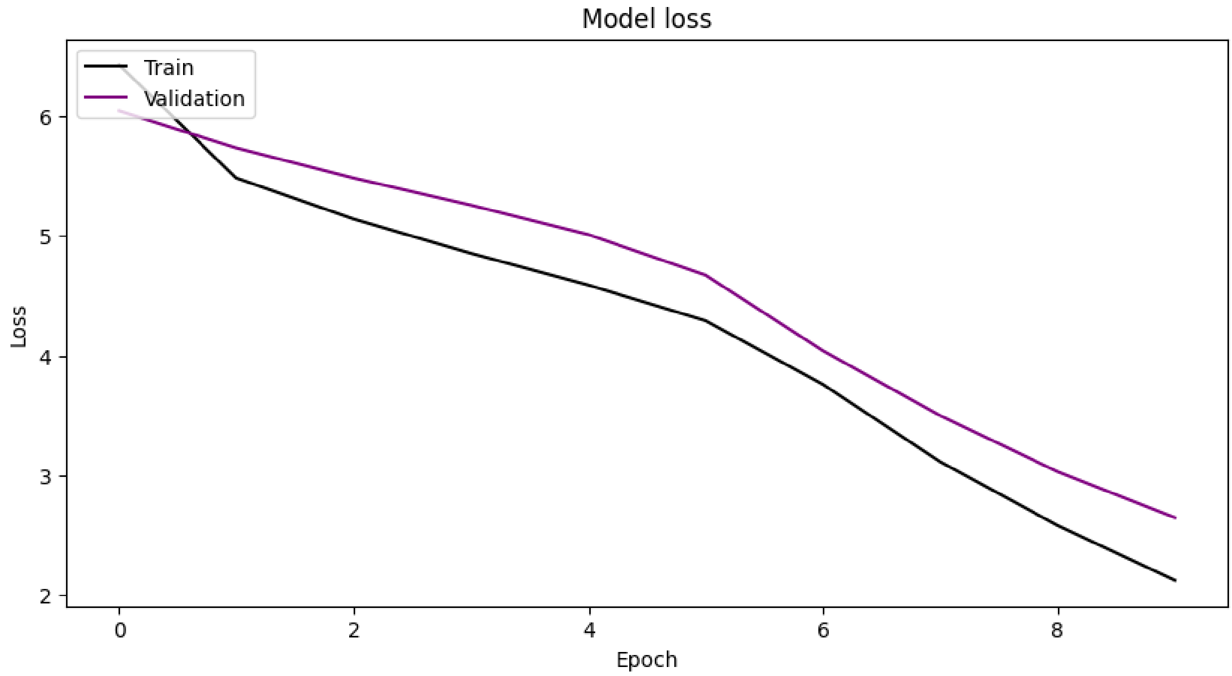


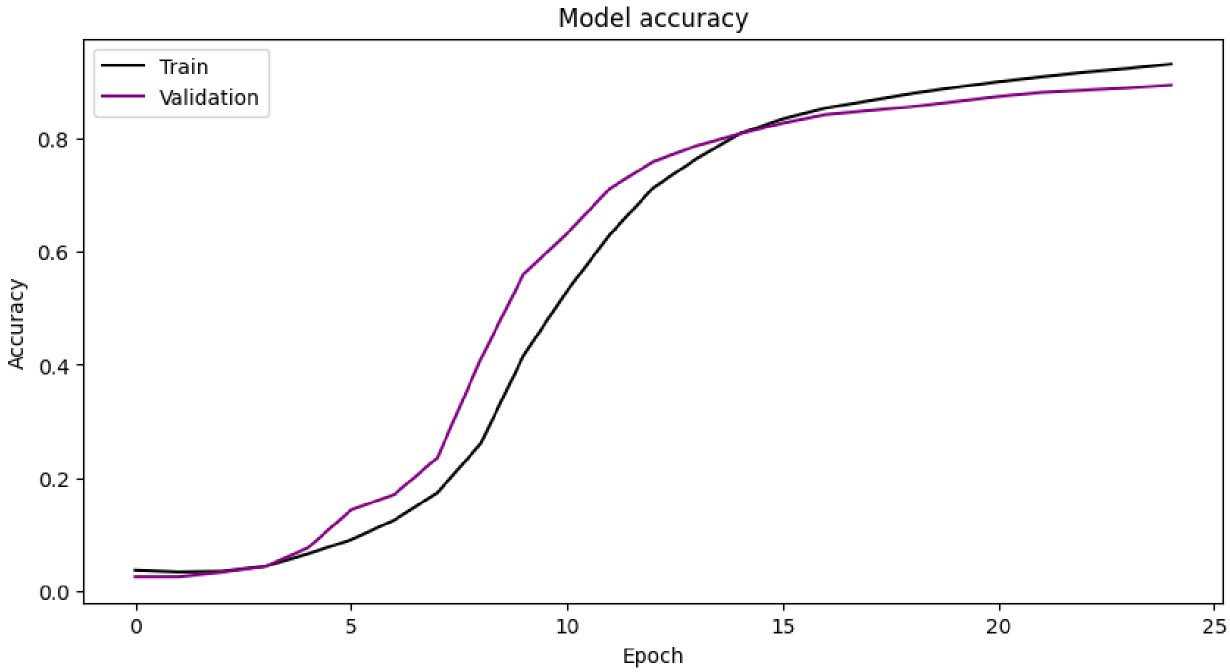
Figure3.9 : Accuracy and error of the LSTM model with a number of epoch = 10.

It seems like the model is performing well but the accuracy is low.

- Testing with number of epochs = 25

Epochs number	Training		Test	
	loss	accuracy	vall_loss	vall_accuracy
25	0.48	0.93	1.39	0.89

Table 3 : Result of training and testing with epoche=25.



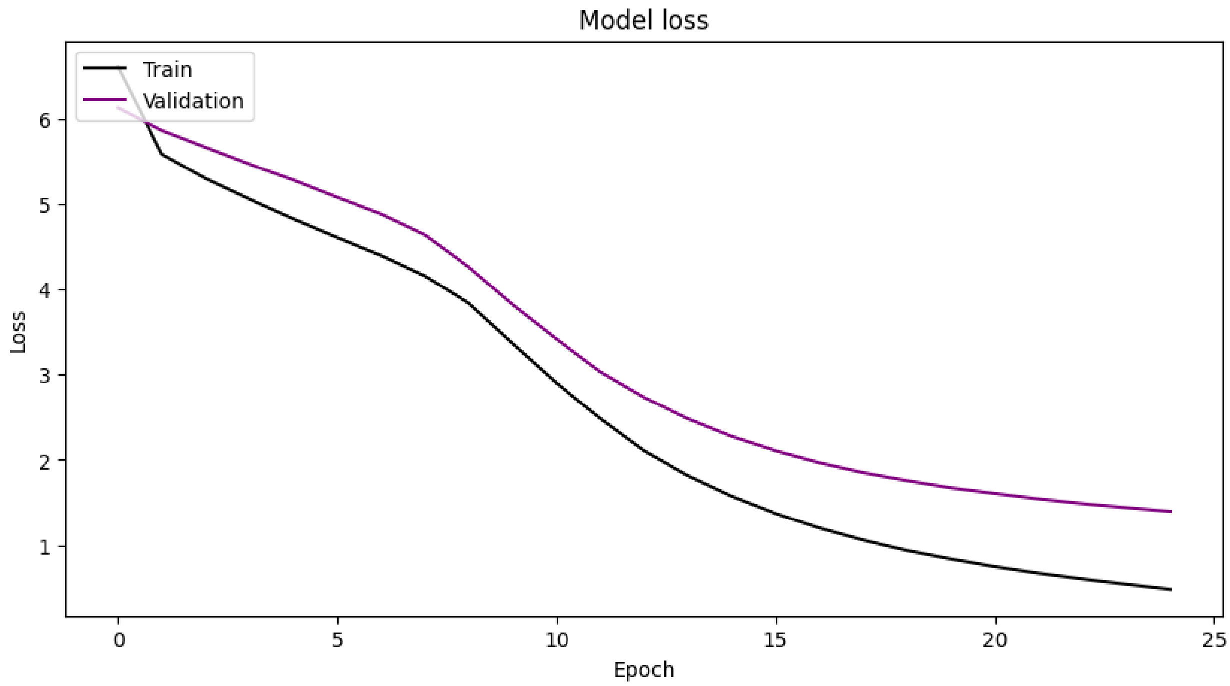


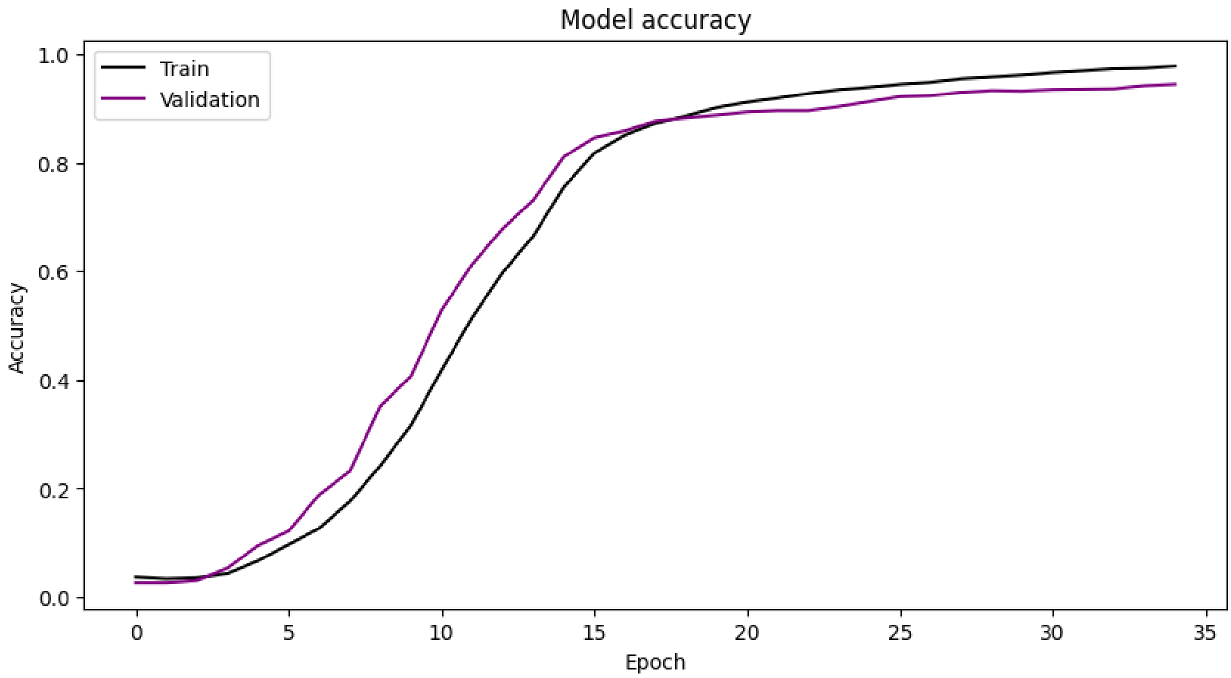
Figure3.10 : Accuracy and error of the LSTM model with a number of epoch = 25.

From Figure 3.10 the accuracy of learning and validation increases with the number of epochs. We observe the same thing for the learning and validation error decreases with the number of epochs.

- Testing with number of epochs = 35

Epochs number	Training		Test	
	loss	accuracy	vall_loss	vall_accuracy
35	0.15	0.97	0.96	0.94

Table 4 : Result of training and testing with epoche=35.



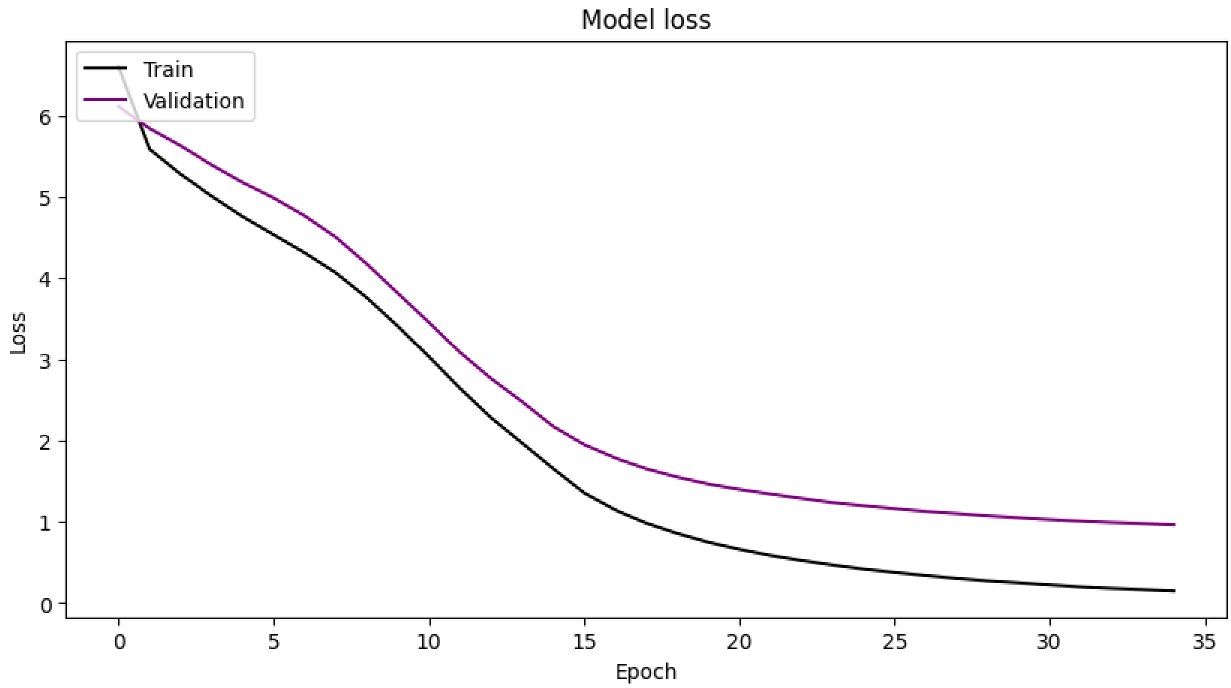


Figure3.11 : Accuracy and error of the LSTM model with a number of epoch = 35.

Great to hear that achieved a high accuracy of 0.97 in Epoch 35 with a low loss of 0.15 in training set. Furthermore, your validation accuracy is also high at 0.96 with a low validation loss of 0.94, indicating that our model is performing well on both the training and validation data. This is a good sign that our model is generalizing well to unseen data and can be used for predicting new data.

- Comparison of the obtained results:

Epochs number	Training		Test	
	loss	accuracy	vall_loss	vall_accuracy
10	2.12	0.71	2.64	0.73
25	0.48	0.93	1.39	0.89
35	0.15	0.97	0.96	0.94

Table 5 : Comparaison results obtained by training our model with different numbers of epochs.

The table shows the architecture number of layers used because it yielded better results in terms of accuracy and lossin number of epoch 35 .We notice, that each time we increase the number of epochs, the accuracy rate increases and the error rate decreases, we notice also that this is not proportional because arrived at a certain threshold, it begins to stabilize and the increase is not as large as at the beginning. In general, reccurent neural network is important and deep, gives good results and the performance of our network degrade if we choose the misfitted parameters.

3.5. CNN MODEL CONFIGURATION AND NUMBER OF PARAMETER

```
 Layer (type)                Output Shape              Param #
=================================================================
 input_16 (InputLayer)       [(None, 100)]             0

 embedding_15 (Embedding)    (None, 100, 100)          1636200

 reshape_15 (Reshape)        (None, 100, 100, 1)       0

 conv2d_48 (Conv2D)          (None, 98, 1, 256)        77056

 conv2d_49 (Conv2D)          (None, 97, 1, 256)        102656

 conv2d_50 (Conv2D)          (None, 96, 1, 256)        128256

 max_pooling2d_46 (MaxPooli  (None, 1, 1, 256)         0
 ng2D)

 max_pooling2d_47 (MaxPooli  (None, 1, 1, 256)         0
 ng2D)

 max_pooling2d_48 (MaxPooli  (None, 1, 1, 256)         0
 ng2D)

 concat_layer (Concatenate)  (None, 1, 1, 768)         0


 flatten_layer (Flatten)     (None, 768)               0

 dropout_layer (Dropout)     (None, 768)               0

 dense_1 (Dense)             (None, 128)               98432

 output_layer (Dense)        (None, 29)                3741

=================================================================
Total params: 2046341 (7.81 MB)
```

Figure 3.13 : The architecture of model 2.

- Result of Model 1:

```
80/80 [==============================] - 152s 2s/step - loss: 1.4071 - accuracy: 0.5778 - val_loss: 1.5891 - val_accuracy: 0.5365
Epoch 8/20
80/80 [==============================] - 162s 2s/step - loss: 1.3323 - accuracy: 0.6037 - val_loss: 1.5285 - val_accuracy: 0.5522
Epoch 9/20
80/80 [==============================] - 174s 2s/step - loss: 1.2692 - accuracy: 0.6227 - val_loss: 1.4920 - val_accuracy: 0.5642
Epoch 10/20
80/80 [==============================] - 170s 2s/step - loss: 1.2180 - accuracy: 0.6390 - val_loss: 1.4705 - val_accuracy: 0.5677
Epoch 11/20
80/80 [==============================] - 163s 2s/step - loss: 1.1753 - accuracy: 0.6517 - val_loss: 1.4467 - val_accuracy: 0.5693
Epoch 12/20
80/80 [==============================] - 158s 2s/step - loss: 1.1353 - accuracy: 0.6615 - val_loss: 1.4179 - val_accuracy: 0.5765
Epoch 13/20
80/80 [==============================] - 158s 2s/step - loss: 1.0992 - accuracy: 0.6727 - val_loss: 1.4043 - val_accuracy: 0.5770
Epoch 14/20
80/80 [==============================] - 163s 2s/step - loss: 1.0636 - accuracy: 0.6824 - val_loss: 1.3982 - val_accuracy: 0.5748
Epoch 15/20
80/80 [==============================] - 158s 2s/step - loss: 1.0296 - accuracy: 0.6928 - val_loss: 1.3886 - val_accuracy: 0.5777
Epoch 16/20
80/80 [==============================] - 153s 2s/step - loss: 1.0001 - accuracy: 0.7020 - val_loss: 1.4090 - val_accuracy: 0.5750
Epoch 17/20
80/80 [==============================] - 159s 2s/step - loss: 0.9678 - accuracy: 0.7090 - val_loss: 1.4003 - val_accuracy: 0.5772
Epoch 18/20
80/80 [==============================] - 155s 2s/step - loss: 0.9384 - accuracy: 0.7186 - val_loss: 1.3984 - val_accuracy: 0.5747
Epoch 19/20
80/80 [==============================] - 165s 2s/step - loss: 0.9078 - accuracy: 0.7289 - val_loss: 1.4035 - val_accuracy: 0.5768
Epoch 20/20
80/80 [==============================] - 165s 2s/step - loss: 0.8782 - accuracy: 0.7396 - val_loss: 1.4096 - val_accuracy: 0.5763
```

Figure 3.14 : The architecture of model 1.

The first layer is an embedding layer, with a dimension of 100 and a vocabulary of 1,000 words. This means that each input word is converted into a vector of 100 real numbers.next, three 1D convolutional layers These layers apply a convolutional filter to the input data to extract local features. The parameters of these layers are filter size, number of filters and activation function. 3

for all layers, the number of filters is 64 for the first two layers and 30 for the last layer. The activation function is ReLU for all layers.the following layers are 1D pooling (max pooling1d) to reduce the data dimension by selecting the maximum value in a sliding window. the window size is 2 and the step is 1. The next layers are dropout layers, for randomly deleting a certain number of neurons during training to avoid overlearning.with a rate of 0.5 for both.The last layer is a dense layer, projecting the input data into a space of dimension 30, which corresponds to the number of classification classes. The parameters of this layer are the number of neurons is 30 and the activation function is softmax ,and the Total number of trainable parameters 2,741,302.

- Testing with number of epochs = 10

Epochs number	Training		Test	
	loss	accuracy	vall_loss	vall_accuracy
10	0.87	0.73	1.40	0..57

Table 6 : Result of training and testing with epoche=10.

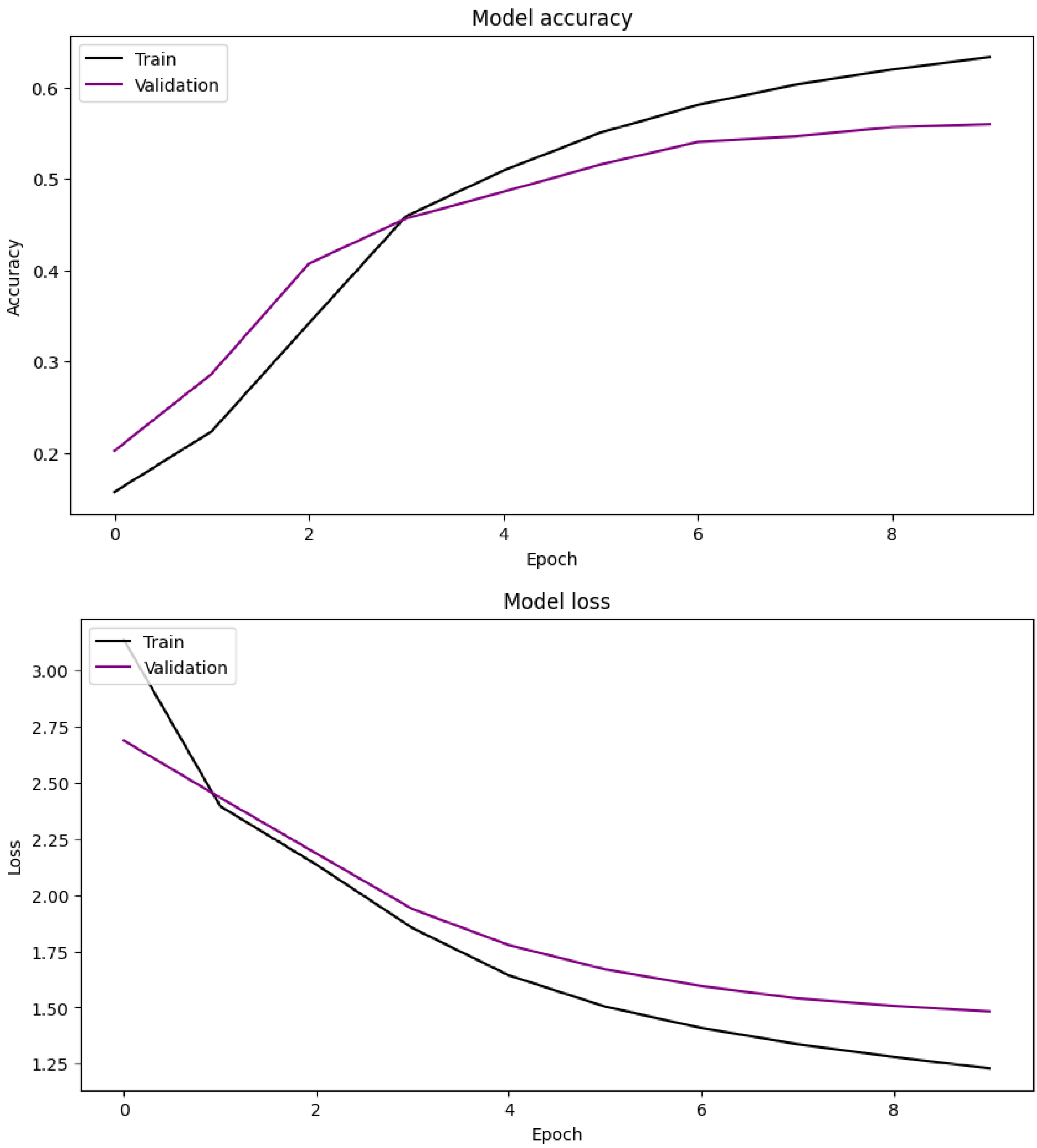


Figure3.15 : Accuracy and error of the CNN model with a number of epoch = 10.

From Figure 3.15 the accuracy of learning and validation increases with the number of epochs. We observe the same thing for the learning and validation error decreases with the number of epochs.With regard to the model's performance over 10 epochs, the results indicate a training loss of 0.87 and a training accuracy of 0.73, suggesting that the model fits the training data relatively well. However, the results on the test set are less satisfactory, with a validation loss of 1.40 and a validation accuracy of 0.57. These figures indicate an overfitting problem, where the model performs well on the training data but fails to generalise to the new test data. Techniques such as regularisation, data augmentation or dropout could be considered to improve the model's performance on the test data.

4. CONCLUSION

In this chapter, we displayed the implementation part of our thesis. We placed our models in function and brought those results for discussion. First, we presented the tools we used in creating the model. Python is our principal programming language, along with the libraries applied to make the execution easier such as Numpy, tenserflow, keras, etc. Then we began showing the initialization of our workspace, with other methods used to resolve the problem. We then observed and discussed the result.

In the next section, we will review what we have done thus far, the chapters, the implementation, the result, etc., and share our future plans and perspectives.

CONCLUSION

AND

PERSPECTIVES

As conclusions this research project explored in depth the possibilities offered by e-learning and artificial intelligence to improve learning processes and meet contemporary educational needs. From the analysis of the state of the art to the design and implementation of the e-learning platform, a number of challenges were addressed and innovative solutions proposed.

We explained our choice in three chapters: the first is where we presented a review of the efforts of the researchers to solve this same problem with similar methods. Afterward, we conducted a comparison between the chosen techniques. These methods offer possibilities for personalizing learning paths and have a significant impact on education. However, it is important to note that each method has specific strengths and weaknesses, underlining the importance of choosing artificial intelligence approaches wisely according to pedagogical objectives.

The second chapter is where we provided an illustrated explanation of our system and how it functions, with a comprehensive explanation of the chosen technique.With a particular focus on the integration of recurrent neural networks (RNN) and short-term memory (LSTM) algorithms specifically. Exploring these algorithms in the context of e-learning highlights the quest for efficiency and quality of search in educational platforms. By delving into the application in dynamic e-learning environments, the study aims to enhance search efficiency and quality for learners navigating through massive amounts of information.

The use of advanced algorithms, such as convolutional neural networks, demonstrates the effectiveness of neural network architectures in dealing with classification tasks in the field of information technology, where they contribute to improving the efficiency and quality of educational content and better adapting it to users' requirements.

The last chapter is where we discuss the implementation part. We presented an illustrated model step by step of our system in action. To make this possible we used PYTHON as the primary programming language for its straightforward, easy-to-learn syntax that highlights readability and, therefore. Several libraries were used for instance the keras tensorflow.

Efforts have been made to ensure a user-friendly experience, optimal accessibility and functionality tailored to the needs of students.despite the progress that has been made, it is important to know the limitations of this work, especially regarding some functions that could be improved or aspects that need to be explored more thoroughly. These limitations provide improvement paths for future work in the field of e-learning and AI applied to education.

By envisioning the future, interesting perspectives integrating new educational technologies and exploring cutting-edge AI methods to improve the personalization of learning pathways. These research paths open the way to new opportunities to improve the effectiveness of distance education systems and contribute to the ongoing evolution of digital education.

Exploring these intriguing paradigms has deepened my passion for the field, a passion that I am eager to continue nurturing as I progress in my studies.

BIBLIOGRAPHY

[1] [3] [7] A. Muniasamy & A. Alasiry. Deep Learning: "The Impact on Future eLearning". *In international Journal of Emerging Technologies in Learning*, 15(1), 188-199, 2020

[2] T. Prudhomme and M. Echeverria, H.Fahad . "An Outlook for Deep Learning in Ecosystem Science. Ecosystems", 24(8), 1825–1836, 2021.

[4] A. Naim." E-Learning Engagement through Convolution Neural Networks in Business Education". *European Journal of Innovation in Nonformal Education (EJINE)*, 2(2), 497-498. ISSN: 2795-8612, 2022

[5] [16] Z. Trabelsi F. Alnajjar , M.M.AParambil, M.Gochoo, & L. Ali .Real-Time Attention Monitoring System for Classroom:" A Deep Learning Approach for Student's Behavior Recognition". *Big Data Cogn. Comput.* 7, 48, 2023.

[6] Y. Hu, Z. Jiang and K. Zhu."An Optimized CNN Model for Engagement Recognition in an E-Learning Environment". *Applied Sciences*, 12(16), 8007, 2022.

[8] M.Alam, ." Codes in matlab for training artificial neural network using particle swarm optimization.2016

[9] P.Halachev. "Prediction of e-Learning Efficiency by Neural Networks". *Cybernetics and Information Technologies*, 12, 98-110,2012.

[10] J. Melesko, and E. Kurilovas."Semantic Technologies in e-Learning". *In 8th International Conference on Web Intelligence, Mining and Semantics (WIMS'18)* (pp. 1-8), 2018.

[11] M. Murshed,M.A.A. Dewan, F.Lin, and D. Wen." Engagement Detection in E-Learning Environments Using Convolutional Neural Networks". *In Proceedings of the International Conference on Dependable, Autonomic and Secure Computing* .(pp. 80-87).IEEE, 2019.

[12] A.Naim, "E-Learning Engagement through Convolution Neural Networks in Business Education". *European Journal of Innovation in Nonformal Education,* 2(2), 497-498, 2022.

[13] P. Jansson, "Single-word speech recognition with Convolutional Neural Networks on raw waveforms," 2018.

[14] G.Li, F.Liu, Y. Wang, Y.Guo, L.Xiao, & L.ZhuA." Convolutional Neural Network (CNN) Based Approach for the Recognition and Evaluation of Classroom Teaching Behavior". *Scientific Programming*, 2021, 6336773, 2021.

[15] A. Muniasamy & A. Alasiry. Deep Learning: "The Impact on Future eLearning". *In international Journal of Emerging Technologies in Learning*, 15(1), 188-199, 2020

[17] [19] P.-S. Chiu, J.-W. Chang, M.-C. Lee, C.-H. Chen, and D.-S. Lee, "Enabling Intelligent Environment by the Design of Emotionally Aware Virtual Assistant: A Case of Smart Campus," *IEEE Access*, vol. 8, 2020.

[18] G.Perry, R.Seidl, A.Bellve, and W.Rammer.An outlook for deep learning in ecosystem science. *Ecosystems*, 25:1–19, 2022.

[20] L.El Youbi El Idrissi., I.Akharraz, and A.Ahaitouf. " Personalized E-Learning Recommender System Based on Autoencoders" . *Applied System Innovation*, 6, 102,2023.

[21] [41] 8 Y. Lia, & L.Chen(Year)." Improved LSTM data analysis system for IoT-based smart classroom". *Journal of Intelligent and Fuzzy Systems*.

[22] N. A. Debbagh and R. Baghdadi. "Optimisation d'Opérateurs de Deep Learning dans Tiramisu (Sparse Neural Networks and Recurrent Neural Networks)" [Undergraduate thesis, Ecole Nationale Supérieure d'Informatique], 2020.

[23] S.Bouktif, A.Fiaz, A.Ouni, & M. A.Serhani . " Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting." *Journal of Energies*, 13(2), 391, 2020.

[24] H.Anantharaman, A.Mubarak, and B. T.Shobana."Modelling an Adaptive e-Learning System Using LSTM and Random Forest Classification". *IEEE Conference on e-Learning, e-Management and e-Services* (IC3e) (pp. 29-36), 2018.

[25] Young-Sang Jeong et Nam-Wook Cho."Evaluation of e-learners' concentration using recurrent neural networks"*The Journal of Supercomputing* .79:4146–4163,2023.

[26] He, Yanbai and Chen, Rui and Li, Xinya and Hao, Chuanyan and Liu, Sijiang and Zhang, Gangyao and Bo. Jiang, " Online At-Risk Student Identification using RNN-GRU Joint Neural Networks." *Journal Information*. 11. 474, 2020.

[27] Z.Shou, M.Xie, J. Mo, H.Zhang . Predicting Student Performance in Online Learning: A Multidimensional Time-Series Data Analysis Approach. Applied Sciences, 14(6), 2522.2024.

[28][32] Killian Janod." La représentation des documents par réseaux de neurones pour la compréhension de documents parlés". *Intelligence artificielle [cs.AI]. Thèse,Université d'Avignon,Français. ffNNT* , 2017.

[29] K. Janod. "La représentation des documents par réseaux de neurones pour la compréhension de documents parlés". *Intelligence artificielle [cs.AI]*. Université d'Avignon, 2017.

[30]Laisen Nie, Xiaojie Wang, Houbing Song, Dingde Jiang, Liangtian Wan, Shui Yu. "Network Traffic Prediction Based on Deep Belief Network and Spatiotemporal Compressive Sensing in Wireless Mesh Backbone Networks." *Wireless Communications and Mobile Computing*, page 4 , 4 janvier 2018.

[31] Y.Kim, T.Soyata, and R.Feyzi Behnagh. "Towards Emotionally Aware AI Smart Classroom: Current Issues and Directions for Engineering and Education." *IEEE Access*, 6, 5308-5319, 2018.

[33] M. Heydarzadeh, S. Hedayati Kia,M. Nourani,H. Henao and Capolino. "Gear Fault Diagnosis Using Discrete Wavelet Transform and Deep Neural Networks". *In 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON)* (pp. 3489-3494), 2016.

[34] C.Lin, Q.Chang, and X.Li, ." A deep learning approach for mimo-noma downlink signal detection". *Sensors*, 19:2526, 2019.

[35] Alberto Pacheco, Ever Flores, Raúl Sánchez ,"Smart Classrooms Aided by Deep Neural Networks *on Mobile Devices* ".*In Inference on Mobile Device*s,Mai 2018.

[36] H.A. Afan , A.I.A. Osman, Y.Essam., AN. Ahmed, Y.F. Huang, O. Kisi, M. Sherif, A.Sefelnasr,K.W. Chau, and A.El-Shafie. "Modeling the fluctuations of groundwater level by employing ensemble deep learning techniques." *Engineering Applications of Computational Fluid Mechanics*, 15(1), 1420-1439. .2021.

[37] F. Kamalov , H. Sulieman , D. Santandreu Calonge ."Machine learning based approach to exam cheating detection". *Journal of PLoS ONE* 16(8): e0254340, 2021.

[38] R.Castro, A.Cousseau, &E. San Martín. "Using an MLP to forecast students' performance in an online course".

[39] C.Iwendi , J. H.Anajemb, & C. J. Anajemba, " MLP-based cheating detection method for multiple-choice online exams".

[40] A.Chanaa, N.El Faddouli "Deep learning for a smart e-learning system." *In proceedings of the International Conference on Sustainable Digital Education (ICSDE'18)*, Rabat, Morocco, October 18–20, 2018.

[42] Tieyuan Liu, Qiong Wu, Liang Chang, Tianlong Gu, "A review of deep learning-based recommender system in e-learning environments", *Artificial Intelligence* .Décembre 2022.

[43] Wang, X., Zhang, Y., Li, Y., & Zhang, D. (2020). A deep learning-based framework for improving accuracy and scalability of recommendations. *Journal of Machine Learning Research, 21*(45), 1-20.

[44] S.Shashi, A.Singh1 & A. Kumar Gupta2 .A Deep Neural Network (DNN) Approach for Recommendation.*Advances in Computational Intelligence and Communication Technology*. 06 avril 2022

[45] P. K.Balasamy, & K.Athiyappagounder. "An Optimized Feature Selection Method for E-Learning Recommender System Using Deep Neural Network based on Multilayer Perceptron". *In International Journal of Intelligent Engineering and Systems*, 15(5), 461-462, 2022.

[46][49] F. Elghibari, RElouahbi and F. Elkhoukhi, "Modèle d’un processus d'apprentissage adapté aux évolutions cognitives du profil de l'apprenant ".*1ère Edition du Workshop International sur les Approches Pédagogiques & E-Learning.*

[47][51],[56] Manju Bhaskar, Minu M Das, Dr. T. Chithralekha and Dr. S. Sivasatya. "Genetic Algorithm Based Adaptive Learning Scheme Generation For Context Aware E-Learning" *in International Journal on Computer Science and Engineering*, 2010.

[48] B.Benhala, A.Ahaitouf, M.Fakhfakh, A.Mechaqrane and B.Benlahbib. "Optimal Analog Circuit Sizing via Ant Colony Optimization Technique". *International Journal of Computer Science and Network Security*, 11(6), 223-224, 2011.

[50] Valigiani, G., Lutton, E., Fonlupt, C., & Collet, P. (2007)."Optimisation par « hommilière » de chemins pédagogiques pour un logiciel d'e-learning.". *Techniques et sciences informatiques*, pages1245-1268,26/10/2007.

[52] S.Bhombe." Swarm Intelligence: The Nature-Inspired Artificial Intelligence Which Uses both Supervised and Unsupervised Learning".

[53]Nicolas Barnier, Pascal Brisset. Optimisation par algorithme génétique sous contraintes. Revue des Sciences et Technologies de l'Information - Série TSI : Technique et Science Informatiques, 1999, 18 (1), pp 1-29.

[54]Thede, Scott. "An introduction to genetic algorithms". *Journal of Computing Sciences in Colleges*. 20.

[55] N. Bigdeli, "The design of a non-minimal state space fractional-order predictive functional controller for fractional systems of arbitrary order," *Journal of Process Control*, 29, pp. 45-56, 2015.

[57] O. Iparraguirre, V. Guevara-Ponce, D. Ruiz-Alvarado, S. Beltozar, F. Sierra-Liñan, J. Zapata-Paulini & M. Cabanillas. *Indonesian Journal of Electrical Engineering and Computer Science.Text prediction recurrent neural networks using long short-term memory-dropout*.29, 3 ,Pages : 1758-1768.2023.

[58] Khalid, M. F. Text Prediction Using Machine Learning (Master's thesis, Linköping University). 2021

[59] H.Bansal,"Text Generation Using LSTM." *Medium*,2020.

[60] S.Soni, S. S. Chouhan, & S. S. Rathore. TextConvoNet: a convolutional neural network based architecture for text classification. *Applied Intelligence*, 53, 14249-14268, 2023.

07-08 /04/2024

[61] "What is Python?," [En ligne]. Available: https://www.educative.io
[62] "Python," [En ligne]. Available: https://www.linkedin.com/

[63] Python. The Python Standard Library. Available: https://python.readthedocs.io

[64] Natural Language Toolkit (NLTK),[En ligne]. Available: https://www.nltk.org/
[65] Plain English. Introduction to NLTK Library in Python. Available: https://python.plainenglish.io/
[66] NLP in Python - NLTK vs. TextBlob,[En ligne]. Available: https://realpython.com/
[67] «NumPy,» [En ligne]. Available: https://numpy.org/.
[68] Introduction to TensorFlow,[En ligne]. Available: https://www.geeksforgeeks.org/

[69] What is Keras?, [En ligne]. Available: https://www.simplilearn.com

[70] Matplotlib,[En ligne]. Available: https://matplotlib.org/

[71] «Difflib», module, Python Documentation, [En ligne]. Available: https://docs.python.org/Available: https://medium.com/

[72] «Elasticsearch», [En ligne]. Available: https://builtin.com/

[73] What is Beautiful Soup?, [En ligne]. Available: https://www.educative.io/
[74] «Html» ,[En ligne]. Available: https://glossaire.infowebmaster.fr/
[75] «CSS», [En ligne]. Available: https://www.atinternet.com/
[76] «JS», [En ligne]. Available: https://developer.mozilla.org/
[77] «Lucidchart», [En ligne]. Available: https://www.lucidchart.com/
[Accés le 07-08 /04/2024].