CONTEXT AWARE RECOMMENDER SYSTEMS USING DEEP NEURAL NETWORK

Thesis submitted in fulfillment for the requirements for the Degree of

DOCTOR OF PHILOSOPHY

By

HARSUMINDER KAUR GILL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT, H.P.

JUNE 2022

Table of Contents

DECLARATION BY THE SCHOLAR

I hereby declare that the work reported in the Ph.D. thesis entitled, "**CONTEXT AWARE RECOMMENDER SYSTEMS USING DEEP NEURAL NETWORK**", submitted at **Jaypee University of Information Technology, Waknaghat, Solan (HP), India** is an authentic record of my work carried out under the supervision **Prof. Dr. Vivek Kumar Sehgal**, Jaypee University of Information Technology, Solan, (HP) India and **Prof. Dr. Anil Kumar Verma**, Thapar Institute of Engineering and Technology, Patiala, Punjab, India. I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of my Ph.D. thesis.

Suminder

Harsuminder Kaur Gill
Enrolment No.: 186214
Computer Science and Engineering
Jaypee University of Information Technology, Waknaghat, Solan
(HP), India
2 June 2022

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled "**CONTEXT AWARE RECOMMENDER SYSTEMS USING DEEP NEURAL NETWORK**" submitted by **Harsuminder Kaur Gill** at **Jaypee University of Information Technology, Waknaghat, Solan (HP), India** is a bonafide record of her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

Vivek Sehgal

Prof. Dr. Vivek Kumar Sehgal
Computer Science and Engineering
Jaypee University of Information
Technology, Waknaghat,
Solan, (HP), India
2 June 2022

Prof. Dr. Anil Kumar Verma
Computer Science and Engineering
Thapar Institute of Engineering
and Technology,
Patiala, Punjab, India
2 June 2022

ACKNOWLEDGEMENT

My time as a PhD Candidate at Jaypee University and Thapar Institute of Engineering and Technology was supported by many people. This page attempts to recognise all those were helpful all the way.

Above all, I thank Waheguru Ji for providing such an opportunity.

I am deeply indebted to my supervisors Prof. Dr. Anil Kumar Verma and Prof. Dr. Vivek Kumar Sehgal who have been a great support throughout the time. They have always motivated me in every situation. Their knowledge and advice have led to the accomplishment of this thesis. Without their guidance this thesis and my professional aptitude would not have been possible.

Further, I am obliged to my parents S. Rashpal Singh and Sd. Manjit Kaur who has always believed in me and supported in my carrier goals. Without them I would have been standing nowhere.

Next, I want to express my gratitude towards my husband Dr. Rajinder Sandhu who has been a continuous source of motivation, support, and guidance. He has always encouraged me towards my carrier goal. He has always helped me to set clear goals and to achieve them. His deep knowledge about the subject was a great benefit for me.

Also, I would also like to express my thankfulness for my in-laws and my son for being such a support.

Further, I want to express my appreciation to every person who directly or indirectly helped me to complete my thesis.

Lastly, Thanks to all members of Computer Science and Engineering Department for providing me all the resources required for the completion of my thesis.

Harsuminder Kaur Gill

(186214)

ABSTRACT

Recommender systems is a class of predictive models that aim at selecting and proposing most relevant items, services or offers to the users based upon their history. These systems aim at enhancing the user's experience based upon what they enjoyed in past. For example, Amazon, Netflix, YouTube, News, etc. As more and more data are being generated, it is the need of the hour to generate user specific recommendations. For example, application suggesting user to purchase items in user's checklist by identifying the location (shopping complex) of the user. These applications generating the recommendations based on the context of the user are known as Context Aware Recommender Systems (CARS). Collecting the context of individual users is not a challenging task as many sensor devices are available at affordable costs. However, analysis of the context, is a complicated task as there may be many repeated, irrelevant, and contradicted context. Many data analysis techniques could be used to analyse the context data such as meta-heuristics algorithms, decision making algorithms and others. The problem with these techniques is high convergence and inference rates, leading to latency issues. Recommendations based on the context is a time sensitive task because if the context changes, the recommendations would be ineffective. Another problem is these techniques perform the context engineering and predictions as two separate tasks. So, in this thesis analysis of contextual and non-contextual data and then making recommendations is proposed to be performed using Deep Neural Networks (DNN). DNNs can perform these tasks as a single entity and with experience these systems perform more accurately. Now, contextual, and non-context data analysis using DNN to make recommendations have several issues that have been addressed in this thesis. Firstly, analysing and removing the irrelevant contexts by dimension reduction while retaining the meaningful properties. Secondly, analysing the contextual and non-contextual data collectively. Thirdly, to identify the contextual relationships and sharing them to other DNNs for effective recommendation generations.

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

In 1994, the Schilit and Theimer in [1] gave the *Context-awareness* phrase in their paper. Since then, it has gained focus in mobile computing, ubiquitous computing, various desktop applications, web applications and smart application research mostly in the last two decades [2]. Multiple definitions and explanations have been listed by various researchers about *context* and *context-awareness*. However, Abowd and Dey's definition in [3] is more relatable and accurate with current smart application developments. They defined context as

"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [3]*."*

Things are usually not black and white; they change based upon the various conditions that affect the situation and preferences [4]. Consider an example, a person may like calm music, when wanted to relax and louder music when need to get motivated. So, previous history cannot help to recommend music to a person due to varying current context of the user, i.e., their mood. To solve this issue, Context Aware Recommender Systems (CARS) were proposed. These systems go one step ahead of traditional Recommender System (RS) by exploiting the contextual information of the user such as time, location, user activity, etc. to understand the situation of the user and its effect on the user preferences, thus, generating personalised recommendations [5]. Contextual data is collected by the components of these applications and after modelling it is used in recommendation generation process. These applications are at hype these days because of the tailored information they present to individual user as per their changing context. For example, if there is an activity in the kitchen during the dinner time, so, a CARS may present recipe-based help depending upon the food available in the house and preferences of the user.

Further, there are many techniques or frameworks that can be used for data analysis in CARS such are heuristic algorithms [6]–[8], rule-based decision frameworks [9] and Deep Neural Networks (DNNs) [10]–[12]. Heuristic algorithms perform better for various optimization problems; however, they have slow inference and convergence rate. These algorithms cannot be applied to make real time recommendations where latency is the major concern. Next comes, rule- based mathematical frameworks that require human intervention to frame rules for decision making which makes it unsuitable to analyse vast, complex and heterogenous data inputs. DNN are self-trained and adaptive models that generates precise and accurate results as more and more data is fed to them [13]. It predicts or recommends based upon past learning and performs better for non-linear or heterogenous inputs [12], [14], [15]. So, it may be proposed that the DNNs are best suitable for data analysis and recommendation generations in CARS.

1.1 Problem Statement

In this era of technology, many sensors are being deployed and consequently a lot of data is being generated. It has been predicted that this number might grow in the future due to advances in the sensor technology, which is making sensors cheaper, powerful, and smaller in size. As billions of these sensors are resulting in the production of large amount of data, it becomes a challenge to store and process this data [16]. However, CARS applications play a vital role here because they help to identify which data is potentially relevant according to various situations. Context in CARS can be used to remove ambiguity and to increase the effectiveness of data analysis.

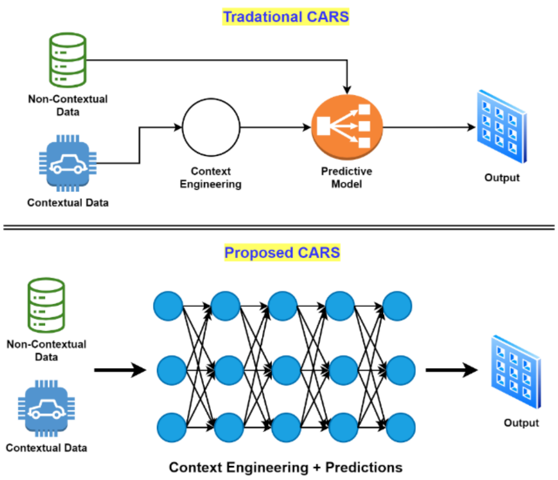


Figure 1.1: Traditional and Proposed Framework for CARS

Further, CARS involve two main activities: context engineering and recommendations generations. To perform these two activities, two different frameworks were used as shown in Figure 1.1. Initially, context engineering is performed on contextual data then along with non-contextual data, it is fed to a predictive model to generate recommendations. In CARS, performing contextual engineering is a crucial task because analysing contextual information and patterns according to the changing context and then evaluating is a challenging task. In this thesis, the use of DNN is proposed that can combine both the activities (context engineering and recommendation generations) and produce accurate results as shown in Figure 1.1. The idea of combining both steps has been conceived from the traditional machine learning algorithms where feature extraction and classification were two separate tasks. DNNs have been extensively adopted and are successfully in performing the feature extraction and classification as a single entity using the complex, vast and heterogenous data. Likewise, it can be used in CARS to perform the context engineering and generating recommendations as a single entity.

1.2 Contributions

This dissertation offers some key contributions in the research of CARS. Firstly, a lot of data is being produced by the sensors and other devices, so, it is the need of time to scale down the dimension of the contextual data, but it should retain the meaningful patterns. It is vital for successful working of CARS to remove and identify the irrelevant or contradicted contextual datapoints. Secondly, in real time applications, both contextual and non-contextual data are collected by the sensors. So, frameworks are required that can generate recommendations using contextual and non-contextual data collectively at the same time. Thirdly, while making real time applications at the large scale, it is crucial to identify the contextual relationships between various intermediate outputs and the sharing of context with other CARS for effective and precise recommendations. All the above-mentioned major findings can be mapped into following research objectives.

i. To transform raw contextual data into low-dimensional space by retaining meaningful properties using Deep Neural Networks.
ii. To analyze contextual and non-contextual data for recommendation generations using Deep Neural Networks.
iii. To design a framework which formulates contextual relationships and its sharing using Deep Neural Network for Context Aware Recommender Systems.

1.3 Thesis Outline

This thesis has been divided into subsequent chapters for better understanding of the work done. Here is the brief overview of each chapter.

Chapter 1 gives introduction about the topic, context, and CARS. It also discusses how CARS are different from traditional RS. It proposes the use of DNN for contextual data modelling and context analysis and reviews that how it is suitable as compared to other techniques. Then, it briefly discusses the problem statement, followed by research objectives of the thesis.

Chapter 2 presents the literature survey to better understand the background of the topic. It represents the explanations and definitions of context, CARS, lifecycle of CARS as proposed by various authors. Then, based upon all the literature studied research gaps in the field as been listed. Also, this chapter explains how DNN has been effectively used for context modelling and context analysis by various authors.

Chapter 3 proposes a framework that converts high dimensional contextual data into low dimensional space using DNN while preserving all the contextual data properties. A use case was taken in which enterprise multimedia file was placed on best suitable storage server. Here, high dimensional contextual data were effectively transformed into lower dimensions without any loss of patterns.

Chapter 4 proposes three different frameworks that could analyse contextual and non-contextual data simultaneously. All the three frameworks are from different domains, hence, showing that this could be applied anywhere irrespective of the field. The first use case taken is to predict irrigation requirements of a crop using various contextual and non-contextual data inputs. The second use case is about predicting and recommending best sleeping posture to an individual based upon the current context of the user. And the third one analyses the contextual and non-contextual data of an epidemic, zika virus. In all the use cases, contextual and non-contextual data was modelled and analysed simultaneously using DNN without additional requirement of context engineering.

Chapter 5 proposes a framework which effectively identifies the contextual relationship between various intermediate results and share them to other CARS to be applicable at large scale. The use case taken was a context aware smart epidemic control, where covid dataset was used.

Chapter 6 highlights all the major contributions of the thesis. It also explains the future scope of the work presented.

CHAPTER 2

LITERATURE SURVEY

People can communicate with each other efficiently due to their underline knowledge of context which is difficult to frame for any human to computer communication [5]. One of the reasons that computers are not intelligent as compared to humans in case of communication is their difficulty to analyse and understand the context of the communication. An artificial intelligence system will work and generate recommendations more efficiently if it can read and understand the current contextual information of its users. So, system can be developed which can collect contextual information and supply it to the predictive system explicitly. However, this is not possible because of the large number of contexts, and it is not possible for a user to know which contextual information is important to which application. On the other hand, smart context aware systems also known as CARS can be developed which can select, collect, analyse, and use contextual information automatically depending on the requirement of any application [17]. This chapter discusses the related work on the CARS which formulates the contextual information in the decision process so that relevant recommendations can be generated.

2.1 Related Work

It is observed from the studied literature that multiple authors have suggested to use context for any recommendation generation. However, the amount of literature found on the context in RS is minimal as compared to the use of RS in multiple domains. So, literature survey has been conducted starting from the background, classification, lifecycle, and usage of contextual information in a RS. This would result in better understanding of the role and relevancy of context in the RS. With the rise of mobile and ubiquitous computing, context has been always a component studied and discussed in various literatures. In this chapter, related work of context aware applications is discussed and its relevancy with RS is provided. For better understanding of context, three separate sections are created in the study of its literature:

i. Classification of contextual data
ii. Usage of Context in RS
iii. Context Aware Smart Applications Lifecycle

2.1.1 Classification of contextual data

Multiple authors have discussed the contextual data classification and their role in any RS. A few important contributions have been discussed in this section of the thesis.

Abowd et al. [3] divided the contextual data into primary and secondary context categories. If the context is directly taken from the sensor by the application, it is called the primary context. On the contrary, if a context is derived from a single or multiple values of primary context, it is known as secondary context. But this kind of classification is ambiguous for any RS development because it is difficult to predict whether a contextual data is primary or secondary. As an example, a home address of any employee can be collected from GPS sensor (primary) or from company's static data (secondary).

Schilit et al. [18] distributed the context into three main categories which are:

i. The geographical location
ii. The nearby contacts
iii. The available nearby resources

The authors suggested that the above said three parameters are enough for any smart application to make effective recommendations. However, issues related to the configuration, deployment, and collection of contexts from various sensors is not addressed in this type of classification.

Henricksen [19] classified the context based upon the configuration and deployment of the sensors which are responsible to collect the contextual data. He divided the context data into four categories which are:

i. Real-time
ii. Static information
iii. Information which changes with less rate
iv. Derived information from above three categories

He discussed the hardware part a lot, however, his classification does not deal with the relationship of contextual information's.

Bunningen et al. [20] provided two higher level categories for the contextual data which are:

i. Conceptual schemes
ii. Operational schemes.

These are two main upper level of categories which further needs subdivision for better allocation to RS.

Then, Perera et al. [16] provided context classification as given by Bunningen et al. in more elaborated fashion. They categorised all the classification given by various authors and classify them into one the category as given by [20]. The classification scheme of contexts in any CARS application given by [16] is as shown in Figure 2.1. The conceptual scheme is further divided into 3 parts:

i. Objective: To understand the related context like where you are, who you are with, what resources are nearby and what is the location.
ii. Entity: To get the information about entity like user, computing, physical, sensor contexts, time, environmental, networking, etc.
iii. Cognitive: It relates with the modelling of cognitive reasoning of the context.

Further, operational schemes are categorized into two categories as deployment and runtime. It provides information about:

i. Runtime and deployment complexities,
ii. The data acquisition price,
iii. Programming framework used.
iv. Tracking of quality, validation, frequency of contextual data update.
v. The processes of accessing the contextual information and context sources e.g. If more data analysis or reasoning is required.

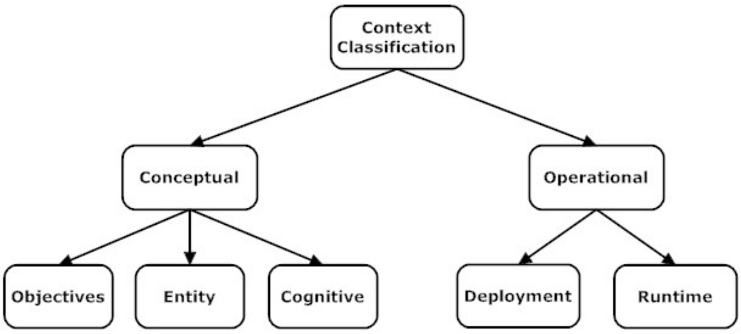


Figure 2.1 Classification of Context

2.1.2 Usage of Context in RS

Any RS designed using contextual data can have multiple features such as to generate a data value, conduct any actions, generate alerts and others. From the studies in [1], [2], [6], [7], the classification of RS features is shown in Figure 2.2.

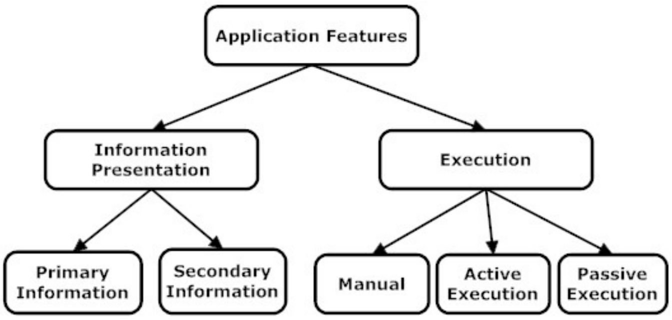


Figure 2.2 Features of IoT based Applications

Any RS which uses context can have two main features which are a) Presenting the information to the user. b) Executing some action based on the context.

Information Presentation:

RSs using multiple sources may collect a large amount of information and as a result generates multiple recommendations. A recommendation provided to the user after the required time is as worst as a incorrect recommendation. Contextual data can be useful in this regard that it can filter the information which is needed to show to the user. For example, an alert for setting an alarm is better suited during the bedtime as compared to any random time in the day. Presentation of the information to the user is divided into two main types which are the primary information, for example, sunrise and sunset time prediction and the secondary presentation of recommendations that are formulated from the primary recommendations or data, such that the suggestion to carry an umbrella based on temperature and rainfall.

Execution:

Automatic implementation of an action by the RS is also an important part of CARS. The authors of [5], [7], studied the role and importance of action execution and its characteristics. For an example, a smart phone automatically changes the brightness level based on the ambience light intensity. Current context of an entity is analysed to perform a machine-to-machine communication which conducts these types of smart actions. Based on the literature studied the smart execution by an RS can further be divided into three types which are:

i. Manual: If a user is performing or asking an RS to perform a specified task then any execution conducted by the RS comes under this category. Like, setting the AC temperature by the user himself.
ii. Passive: In this type, the CARS constantly supervises the system and provide a list of actions from which user can choose any. For example, when the user goes for shopping, the system presents him the list of discounts available for various items.
iii. Active: In this type, the CARS automatically performs appropriate actions. For example, if the CARS detects that amount of milk is below a threshold, it will place an order for fresh milk automatically.

2.1.3 CARS Lifecycle

Hynes et al. [8] discussed the importance of context lifecycle by stating that any CARS has two lifecycles which are enterprise lifecycle and context lifecycle. Furthermore, Perera et al. [5] reasoned that the software engineering lifecycles are standardized but the lifecycles of contextual information are at the early phases. A few relevant lifecycles found in the literature are listed in Table 2.1.

Based on the literature studied, the lifecycle of a CARS can be allocated into four distinctive stages shown in Figure 2.3. Each stage is an individual research area and efforts have been made to address these areas and identify the research gaps.

Table 2.1 Lifecycles Proposed for the CARS

Literature	**The Proposed Lifecycles of Context**
Chantzara and Anagnostou[21]	i. Sense ii. process iii. disseminate iv. use.
Ferscha et al.[22]	i. sensing the context from sensor ii. transform the raw context iii. standard representation of context iv. develop the rule base v. actuation of the context
Wrona and Gomez[23]	i. discovery of context ii. acquisition of context iii. reasoning of context
Hynes et al.[24]	i. sense the context data ii. transmit the context data iii. acquire the context iv. classify the context v. handle the classified context vi. disseminate the context vii. use the context in decision viii. delete the context ix. request new context x. maintain the collected context xi. create disposition of collected context
Baldauf et al.[25]	i. deploy the sensors ii. retrieve the context iii. process the context iv. store the context v. use the context in decision making
Perera et al. [16]	i. Acquisition of Context ii. Modelling of Context iii. Reasoning of Context iv. Dissemination of Context

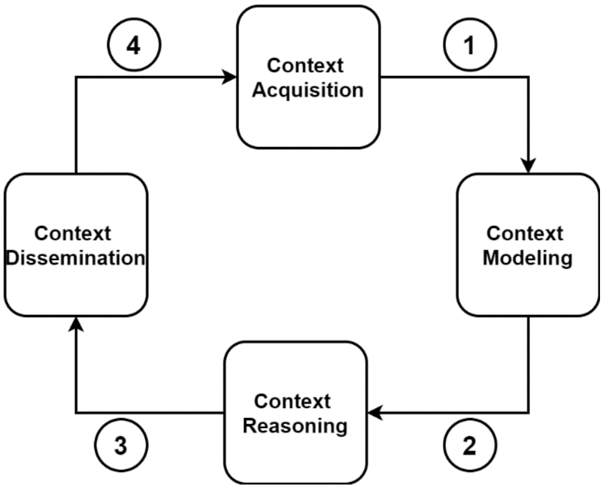


Figure 2.3 The Most Suited CARS Lifecycle

Context acquisition:

The acquisition of the context is one of the fundamental stages in the development of any CARS. There is no role of subsequent stages if the wrong or no contextual data is collected. Factors influencing the contextual data acquisition are [5], [13]:

i. Need of the data: Push and Pull strategies are followed in this case. If the data is required by software application, then it is called pull. If sensor sends data to software application automatically, it is called push.
ii. Frequency of data: Trigger based sensors send data based on some trigger and periodic sensor send data after a pre-specified interval.
iii. Source of data: [26]–[28] provided sources of data in CARS which are direct from the sensor, through a middleware API's and data stored in warehouse.
iv. Process of collection: [16] divided the process of collection of data into three parts which are primary, derived and manual.

Context Modelling

Selection of appropriate modelling technique for acquired context is an important step in any CARS. Authors discussed the context modelling techniques in the relevant literature [17][18] in detail. Based on the studied literature, commonly used context modelling techniques are:

i. Key-Value Modelling: A value is associated with a key of any context data. This type of modelling was used in early mobile systems, and it is not suitable for CARS application because of its scalability and reliability issues with the colossal number of devices deployed.

ii. Markup Modelling: In this type of modelling, a scheme is followed to model the context of any application. XML is the most common used markup [29], [30]. This type of context modelling cannot support CARS applications because the markup file will become large, and it will become complex and time consuming to decode it in the real-time.

iii. Graph based Modelling: To study the hierarchal structure in context modelling, graph based modelling techniques have been proposed by many authors [19], [31], [32]. Graph based modelling can be used in CARS because it can divide the deployed sensors into different concept levels.

iv. Object Based Modelling: This modelling technique works by treating the context of sensor or user in the form of objects. It provides better scalability and usability as compared to previous discussed schemes. This type of modelling can be used for CARS where different devices or sensors (treated as object) can communicate with each other to form any decision. If basic design of CARS is also object based, then sensor integration is also very easy and effective. However, this type of modelling will not be able to take decisions using semantics.

v. Ontology based modelling: In this type of modelling, context of any sensor or user is represented using different ontologies techniques such as OWL, RDF and RDFS. This will provide system many reasoning capabilities to model the context in effective manner. Many author have studied the applicability of ontologies techniques in context modelling [33]–[35]. So, this technique is also suitable for context modelling in CARS applications.

Data Analysis

After context modelling, the data analysis is conducted to mine useful patterns from the collected and modelled contexts [17], [26]–[29]. The related literature of context reasoning and data analysis is substantial and cannot be discussed in detail for the brevity of this thesis. A survey was conducted by Lim and Dey [36] about different data analysis techniques used in many CARS based projects. The results have been shown in Table 2.2 where 109 projects were surveyed from three major conferences over the five years. It can be analysed from the Table 2.2 that more than 50 percent of the researchers used rule-based approach; however, decision tree, naïve bayes and Hidden Markov models were used by only 15 to 13 percent researchers. However, these techniques are capable of handling data with small number of attributes and instances. The amount of contextual and non-contextual data generated these days by many CARS is large that these algorithms would result in high convergence and inference time. Many CARS are time sensitive where the latency of recommendation generation is equally important to the quality of the recommendations [37]. Furthermore, these algorithms require pre-processing and feature extraction of all the attributes to make quality recommendations which is not possible in diverse and heterogenous data collection methods of CARS.

Table 2.2 Machine Learning Algorithms Used by Projects in [5]

Machine Learning Algorithm	**Number of Projects**
Simple Rule Based	58
Decision Tree Algorithms	16
HMM	14
Naïve Bayes Algorithm	14
SVM	5
K-Nearest Neighbor	2

Context Dissemination

After all the stages discussed previously are completed in a CARS lifecycle, final requirement is the dissemination of generated recommendations, so that the required execution is performed. Recommendations generated may be further send to other applications, smart devices, or users. Following two methods can be used to share the generated recommendations [5]:

Query Based: For the query-based CARS, recommendations generated will be sent to the user or the application when they ask for it in the form of query.

Subscribe: In the subscribe based system the recommendations are generated and shared after a fixed interval of time.

2.2 Context in Recommender Systems

In last decade researchers progressively started working towards context-awareness. However, it is evident from [30], [38], [39], and others that in the early phase, focus was on collecting data using smart devices. Hence, the title and subject matter of most of the research articles was on collection, configuration, and deployment of IoT based devices. However, they were completely relevant to any CARS system. Different techniques and methods to model the context were also proposed, and contextual data analysis techniques were discussed in the later period. In this section, some of the notable contributions are discussed.

In 2012, Yanwei et al. [30] argued that context will play an important role in the designing of any smart application. They stated that the effective communication of the context from smart thing to thing can result in better recommendations. A framework was proposed by them to send different contextual information to other devices.

In 2015, Abbas et al. [40] presented a survey on various computational intelligence techniques like fuzzy set, ANN, evolutionary algorithms, however, no framework was proposed.

In 2016, Chen et al. [41] developed a hierarchical context modelling technique which they argued that was better than traditional methods. The limitation was the scalability of the proposed context modelling technique. They also did not discuss about any context analysis technique.

In 2016, Gil et al. [38] surveyed the computing technologies, smart applications, and the context data for the development of any CARS. They also listed some of the open challenges to handle context in an effective way.

In 2016, Sachdeva et al. [42] stated that at the end a machine learning algorithm will analyse context and generate recommendations. So, they compared various available machine learning techniques for analysis of context.

In 2016, Du et al. [43] proposed that context should be analysed in the data plane before passing it to the cloud computing platforms. They developed a software defined network-based prototype to analyse the context in the network itself based on the information provided in the contextual data header. They stated that if CARS is latency sensitive, then context analysis was conducted in the data plane.

In 2016, Gill et al. [39] designed an application for elderly people which uses contextual information to manage the supply chain of products. Based on the availability of the products in the home, it generates alerts and warnings messages for the elders.

In 2016 Amin et al. [44] proposed to develop an extra layer in the layered architecture of IoT applications. This extra layer on top of the device layer analyses the context in its raw form and present it to the upper layers in a standard format. They tested this concept in a use of fault management in electric power distribution which was successful as compared to traditional methods.

In 2016 Khan et al. [45] argued that for better sharing and understanding of the contextual information a unified communication channel should be developed. They tested it in a smart home environment where all sensors use same communication method, it does result in less latency and better understanding of contextual information by different devices.

In 2016 Chen et al. [41] argued that CARS should have an effective method to search for contextual information, if required. They also stated that traditional search algorithms would not be effective for CARS. So, they proposed a semantic based ontology method to search for appropriate context deployed in a smart home setting.

In 2016, Sandhu and Sood [46] created a framework using the game theory to use the value of sensors which are relevant to current context from large pool of sensors based on the current context values in a smart home CARS.

In 2016, Kamienski et al. [47] discussed the role of a unified application development platform with some standard API's for facilitation of fast development of CARS. They proposed a simple development environment and developed a CARS which manages the energy of smart buildings.

In 2016, Rokni and Ghasemzadeh [48] proposed a new method where a smart object can reconfigure its behavior depending on the working and current context of different entities. They discussed the possibility of a smart object to learn from another smart object in the environment and adopt their algorithm.

In 2017, Sassi et al. [49] discussed about the importance of context in RS. They provided the basic idea behind the concept, CARS. Any context modelling or context data analysis technique was not proposed.

In 2017, Paradarami et al. [50] proposed the use of DNN for recommendation generations in RS and data analysis. Both content and collaborative filtering techniques were used; however, context was not used.

In 2017, Kim et al. [51] used the context of words used by the reviewers while writing any review for any item. They used Convolution Neural Network (CNN) model to frame and analyse the contextual information.

In 2018, Liu et al. [52] designed a Neural Network (NN) for the recommendation of idioms based on the context of the essay topic.

In 2018, Villegas et al. [53] surveyed large number of research articles and detailed analysis was provided about how context plays important role in generating recommendations and how this is the future of RS.

In 2018, Katzman et al. [54] used DNN for generating treatment recommendations. Their results show that DNN performed better in identifying complex relations as compared to other techniques. The contextual data was explored for personalised treatment recommendations.

In 2019, Batmaz et al. [55] provided the review of frameworks and relevant literature where recommendation systems uses DNN for making effective predictions. They did not consider context; however, researchers agreed that DNN would also be an ideal solution for large scale CARS.

In 2019, Liu et al. [56] developed a CARS for robots using DNNs. The training time of the model was extremely high due to large dimension of contextual data. They did not perform any dimension reduction in order to decrease training time.

In 2019, Raza and Ding [57] presented a detailed progress of CARS irrespective of the field they were used. They conducted in-depth analysis of algorithms used in RS, how these algorithms used in CARS are different. They highlighted the modifications required on top of RS to generate context-aware recommendations. Finally, the challenges and research opportunities in the field are also listed.

In 2019, Pradeep and Krishnamoorthy [58] presented the survey about three main building components of CARS; the context modelling, context organisation and context middleware. No framework was proposed, a detailed information about three components were presented.

In 2020, Djellai and Adda [59] proposed a DNN model by using Hidden Markov Model and ANNs. The aggregated model shows improved robustness and training accuracy as compared to other benchmarking models.

In 2020, Wang et al. [60] proposed a bidirectional Long Short Term Memory (LSTM) based context-aware citation recommendation model. The results show effective learning of the relationship between author information and citation index.

In 2021, Boppana and Sandhya [61] presented a sentiment based recommendation generation model using DNN. The results show higher accuracy while using DNN as compared to other machine learning models.

2.3 Research Gaps

It was observed that research gaps can be easily classified based on the life cycle of CARS presented in Section 2.1.3. To the best of our knowledge some of the existing gaps in the field of study are presented in this section.

2.3.1 Data Acquisition

Due to increase in the volume of sensors and other smart devices, effective context acquisition is the primary issue in any CARS. Some of the research gaps in context acquisition are:

i. Effective configuration of sensors and devices so that most of the data sensing area and points are covered using minimum number of devices.
ii. To increase the interoperability of CARSs, development of same standards for format and structure of data and context sensed is not available in the relevant literature.
iii. Factors including quality of contextual data needed, cost of installation and complexities of data analysis algorithms are not studied for effective configuration of devices.

2.3.2 Modelling the Context

As studied, modelling the context is very important aspect in development of CARS which can use millions of sensor data available through different API's. Context modelling has very vast literature but modelling the context for billions of sensor devices and thousands of CARS running concurrently is not much explored area. Some of the research gaps in modelling the context for CARS applications are:

i. Discovery of appropriate context for any specified user, application or device is not much explored area. Selection of appropriate attributes, characteristics, relationships should be effectively discovered before making any decisions in any CARS. In today's digital environment the numbers of entities are colossal, so discovery of appropriate context is important research area.
ii. Selection of appropriate context parameters for any specified CARS.
iii. Pre-processing of raw contextual data so that quality predictions can be obtained.

2.3.3 Data Analysis

Data analysis is vastly studied field by research community for CARS. However, following research gaps are identified in this area:

i. Analysis of contextual and non-contextual data for recommendations generation using machine learning algorithms which have high scalability.
ii. Context relationship, context sharing.

2.3.4 Execution/Presentation

Execution and presentation of information to end devices/users is final, yet important step in the proper working of CARS. Following gaps are identified in this area:

i. Standardization of data sharing API's so that information received can be easily decoded.

ii. QoS of the information received.

2.4 Conclusion

From the literature studied and analysed, it can be concluded that DNNs are the way forward for designing and developing CARS. Further, it can also be summarised that any development of CARS should also focus on context lifecycle along with its software lifecycle.

CHAPTER 3

DIMENSION REDUCTION OF CONTEXTUAL DATA USING DNNs

CARS has an edge from the traditional recommender system because they can generate recommendation which are relevant to the user, time, and location [3], [56]. In current era of data collection devices, various sources can be used to collect the value of the context. The issue with the collection of contexts from various sources are:

i. Repeated context data collection
ii. Irrelevant context data collection
iii. Contradicted contexts

However, a CARS application cannot skip any contextual information because it may be irrelevant for most of the users, but it may be relevant for a few users. So, methods or frameworks needs to be devised which can convert heterogenous high dimensional context data into a low dimensional space so that CARS applications can use them in an effective way. This low dimensional data for context should have properties from all the raw contextual information collected and it should be able to use by all the users in the CARS environment.

In this chapter, a CARS has been developed which recommends the storage location of a multimedia file generated from an enterprise based on security, privacy, and context of creation of the file. DNNs has been used to convert raw contextual information of a file into low dimensional space which is then utilize by another DNN for deciding the location. This use case shows the benefit of converting raw contextual information into low dimension while generating any recommendations.

3.1 Introduction

In the era of Industry 4.0 [62], one of the main factor for any organization to be successful is the secure and effective communication of the information [63]. A large amount of information is collected by the Enterprises in two major categories which are multimedia and non-multimedia files where these both categories can have various formats. Both types of files require an appropriate placement to secure the information they carry. However, in the era of information and technology, multimedia information is prominent these days. Every multimedia file in an enterprise has variable policies according to their required privacy, so, a static rule-based method to store multimedia files on the respective file servers may not work effectively. Hence, there is a need to develop effective CARS framework that can locate a server with required level of security and privacy while sharing any file inside and outside of the enterprise. In the current era of data [64], evaluation of stored multimedia files is one of the difficult tasks that enterprise needs to perform, because there may be abundant data to store [65]. Mostly enterprise uses the cloud computing to store large multimedia files for easy access and updating process. So, the proposed framework in this chapter deals with every kind of storage infrastructure which is called computing infrastructure. This computing infrastructure should be accessible to all the employees of the enterprise whether they are inside the enterprise private network or not. However, file storage with appropriate security and privacy is the major challenge with these setups [66]. For example, any confidential information can be easily leaked to outside world, if a sensitive document is accidently placed on a public network where it can be hacked easily. On the other hand, a lot of secure storage server space will be wasted if a public file is hosted on it. Hence, context aware file placement on storage servers is need of the hour for any organization [16]. Enterprise file can be anything like a video, audio, documents, sensor values etc. and there may be large amount of information about each file. So, the objective of this chapter is to propose a framework which converts the various raw contextual information of the file into single or few values and decides the appropriate storage based on its contextual information.

To accomplish the objective, the DNNs are used at two levels. At initial level, the various contextual values of the file are feed to a DNN which generates a single value that is the File Sensitivity Rating (FSR). In this framework, the files are divided into four main classes: general purpose, public, secure, and sensitive. It is not feasible to allocate every file into a hardbound these four categories. So, at level 2 a SOM [67] based DNN is proposed which finds relevancy of a file between all these categories and generate a list which contains probability values of how likely a file could be associated to each category. Next, the FSR and values of SOM probabilities along with some file parameters like explicit security value, format, and size are passed to a Back Propagation Neural Network (BPNN) to finally decides the appropriate storage space to store the enterprise file. The performance evaluation of converting high dimensional contextual data to a single value is conducted using data generated by Trumania [68] package. Results from the experimental setup proved the hypothesis that the conversion of contextual data from high to low dimensional space improved the results and training time.

3.2 Proposed Framework using DNNs

Many DNNs were combined into a single NN that performed the context engineering of various context values and then make prediction about the allocation of a file.

Figure 3.1 shows various elements of the proposed framework that are Context analyser, Multimedia file, Allocation decision and SOM block.

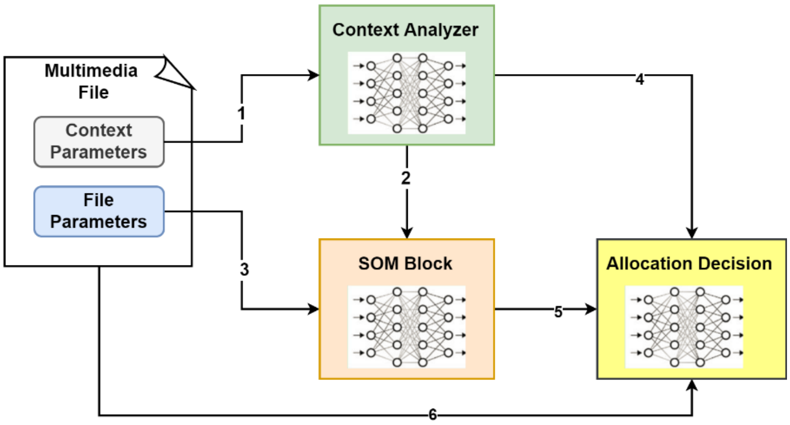


Figure 3.1 The Proposed Framework for Dimension Reduction using DNNs

The input parameters are divided into two classes known as context parameters and file parameters. Context parameters store various contextual data values such as what type of document is being created or shared, who created that file, to whom it is intended to be shared with, when was the file created. All these parameters are automatically generated whenever a new file is created by a user or the system. Next, file parameters are the static properties of the file like format, size, and Explicit Security Requirement (ESR) value. All these parameters are listed in Table 3.1. The context analyzer block takes all the file contextual values and converts the high dimension contextual parameters into a single value called the FSR. FSR value lies between 0 and 1, where 0 is least sensitive file and 1 means most sensitive file. Then, the FSR value is further used to create a list of probability values which represents the likelihood of a file to be sensitive media, public media, secret media, and general-purpose media. These two steps are the context engineering steps which converts various values of contextual parameter into a FSR value and a list of values. Lastly, a simple BPNN is used to make decision related to the allocation of the file on one of the storage servers.

3.2.1 The File

The file in the proposed framework can be any multimedia file which is generated in the organization or shared by someone either inside or outside the organization that may or may not have important information. Example of these files can be internal memo, public disclosure, sensor value, surveillance tapes or any other file which may have a digital footprint. As discussed in the previous section, any file created, generated, or shared in the organization will contain two types of parameters that are the file and contextual.

Table 3.1 lists various inputs of the proposed framework, these parameters are for the reference only and more parameters can be added, if required.

Table 3.1 Parameters of a File Shared/ Created in the Organization

S. No.	Name	Parameter Type	Description
1.	What	Context Parameters	The type of file or basic description of file.
2.	When		The exact time as per GMT.
3.	Where		The exact location (inside/outside of organization).
4.	Who		The username or employee code who generated.
5.	Whom		Who can view the file.
6.	Format	File Parameters	Standard format name.
7.	Size		Memory size requirements of the file.
8.	ESR		The value ranging between 0 to 1.

3.2.2 Context Analyser Block

This block takes the context parameter values and outputs one single value between 0 and 1, known as FSR. One hot encoder is used to encode the contextual values. The overall design of the proposed context analyser block is shown in Figure 3.2 along with the hot encoder values. Every input to the context analyser is in the form of categorical one hot encoder with varying values. For instance,

$\bar{X}_1$= {video file, document file, text file, audio file} is the 'What',

$\bar{X}_2$= {before hours, office hours, after hours} is the 'When',

$\bar{X}_3$={public network, private network, fog environment} is the 'Where',

$\bar{X}_4$={other employee, client, management} is the 'Who',

$\bar{X}_5$={staff, public, client, specific members} is the 'Whom'.

The above given values are the examples, more values can be added if required.

The size of the input is n$*$ e where n are the context parameters containing the e values each, e can be varied from one context parameter to another; however, for easy understandability, it is kept same in the proposed framework. Each value of every context parameter is encoded using one hot encoder [11]. The numeric value for each context is written as X_{pq} where p is the context parameter and q is the identifier for context value. Hidden layer takes the output of the input layer, and it has t units. Input to hidden layer is the dense layer which has the weight matrix W were,

$$\overline{W_q} = (W_{q1}, W_{q2}, ---, W_{qt}) \tag{3.1}$$

$$\bar{h} = (h_1, h_2, ----, h_t) \tag{3.2}$$

$$h_r = \sum_{p=1}^{n}\left[\sum_{q=1}^{e} W_{qr} * x_{pq}\right] \forall r \in \{1, ---, t\} \tag{3.3}$$

In the vector form,

$$\bar{h} = \sum_{p=1}^{n} \sum_{q=1}^{e} \overline{w_q} * \overline{x_{pq}} \tag{3.4}$$

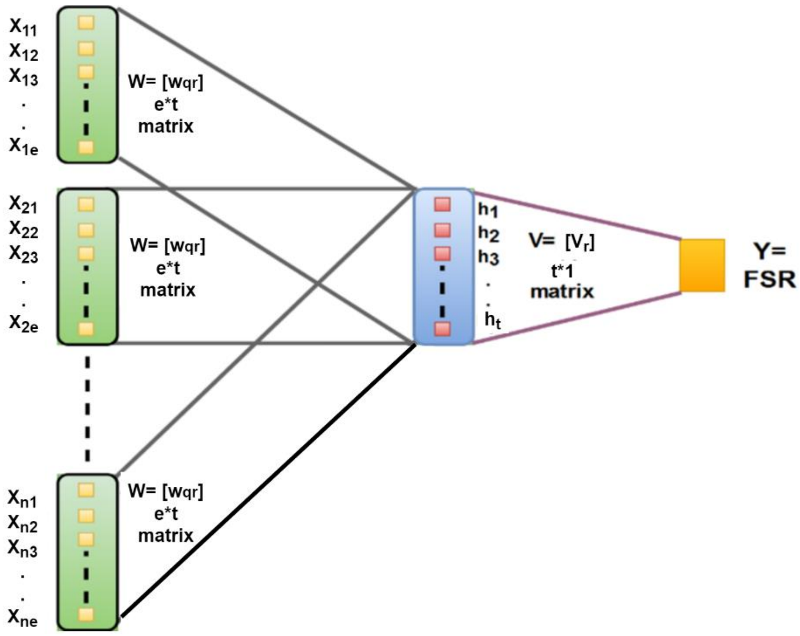


Figure 3.2 Hot Encoder for Context Analyzer

Subsequently, hidden layer is used as input to the final output layer. Weights in the output layer are written as $V = [v_{rq}]$, and is a $t * 1$ matrix. To squash the FSR rating between 0 and 1 the sigmoid activation function is used. The final output y, after applying sigmoid function can be written as:

$$y = \frac{1}{1+e^{(-(\sum_{r=1}^{t}\sum_{q=1}^{e} h_r * v_{rq}))}} \tag{3.5}$$

3.2.3 SOM block

Different file servers can be created on the available storage space with varying capabilities of security and privacy. Every file must be stored on a file server according to its requirements. However, it is difficult for the user or the system to find exact file server to store the shared or created file. The file can lie somewhere between any of the four categories instead of being allocated to just one category. So, the proposed framework uses SOM [69] algorithm to find the probabilities of the file being allocated to one of the four categories which are public, secret, sensitive and general purpose. This algorithm generates a list having probability of being allocated to every category. This is again the context engineering step where various context and file parameters are used to generate these probability values. The input used in the SOM algorithm is the ESR, size, format, FSR value specified with the file. Figure 3.3 below shows pictorial representation of SOM algorithm used in the proposed framework and Algorithm 3.1 explains it in detail.

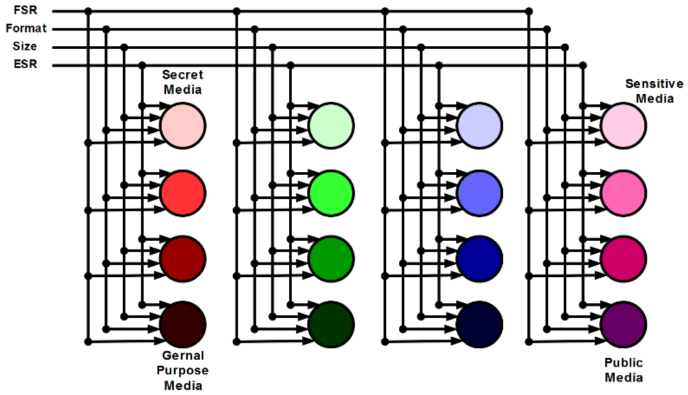


Figure 3.3 SOM Cluster for Finding the Probability Values

A SOM based lattice as shown in Figure 3.3 is proposed to be created of the file servers in the storage space. In the lattice the storage servers are virtually placed in a certain pattern. For example, in the Figure 3.3, top left hand side corner has file servers which are strictly for secret file storage. Similarly, top right-hand file servers are for the sensitive files. If the file requirement based on the generated probabilities values is something in the middle of the secret file and the sensitive file, then that file can be placed on the storage file servers which are in the middle of these two types on the virtual lattice of file servers. SOM algorithm will identify the best suitable file server, or its best closet file server based on the inputs. This output will be provided to the allocation block which will select the possible available file servers to store the file.

Algorithm 3.1: Creation of Virtual Server Lattice using SOM algorithm

Step -1: Initialization

Use the random number generator to initialize the weights of SOM algorithm with small random values.

Step-2: Sampling

To draw a sample of input vector X with certain probability.

Let p denote the dimension of input space vector Z (here p= 4). Therefore, input space can be represented as

$$Z = [z_1, z_2, - - --, Z_p] \tag{3.6}$$

The synaptic weight vector W will have same dimension as of input vector. Therefore,

$$W_j = [w_{j1}, w_{j2}, - - - - -, w_{jp}] \tag{3.7}$$

Where j ranges from 1 to n and n is the cardinality of neurons in the proposed network. In the Figure 3.3 the $n = 4 * 4 = 16$.

Step-3: Find the similar matches

In this step, the minimum Euclidean distance of input Z and weights W are used to select the best matching neuron. Euclidean distance is calculated for all the neurons for every input pattern based on which best matching neurons are selected. The neuron which satisfies the below condition is declared winner of the competition.

$$b(z) = \arg\min ||z(i) - w_j|| \tag{3.8}$$

Where $j \in A$, A is the lattice of multimedia file, $b(z)$ is the index of winning neuron that matches the z input.

Step-4: Updating

In this step, synaptic weight vector is adjusted for all the neurons that are connected to winning neuron in the network using equation 3.9

$$w_j(i+1) = w_j(i) + \eta(i)h_{j,b(z)}(i)(z(i) - w_j(i)) \tag{3.9}$$

Where,

$\eta(i)$ is the learning rate of SOM algorithm, $h_{j,b(z)}$ is the neighbourhood function which is centred around the best neuron $b(z)$ which can be written as:

$$h_{j,b(z)} = e^{\frac{d_{j,b}^2}{2\sigma^2}} \tag{3.10}$$

Where $d_{j,b}$ denotes the lateral distance of the excited neuron j and best neuron b calculated as

$$d_{j,b}^2 = \|r_j - r_b\|^2 \tag{3.11}$$

Where r_b is the position of best neuron b and r_j is the exact location of selected neuron j. If time instant i is added to the neighborhood function, then it can be written as

$$h_{j,b(z)}(i) = e^{\frac{d_{j,b}^2}{2\sigma^2(i)}} \tag{3.12}$$

Where $\sigma(i) = \sigma_0 e^{-\frac{i}{\tau}}$, σ_0 is the value of initialized σ, τ is the time constant. Both $\eta(i)$ and $h_{j,b(z)}(i)$ are changed automatically during learning for best outcome.

Step-5: Repetition

Repeat the Step-2 until updation is negligible or no visible changes in the feature map are seen.

3.2.4 Allocation Decision

So, after the conversion of all raw contextual data to a FSR value by the context analyzer block and a probability of file servers that can be allocated by SOM block, along with other file parameters and available file servers are feed to a BPNN. It outputs the appropriate server to store the file. Figure 3.4 shows the architecture of the used BPNN with all its inputs and output.

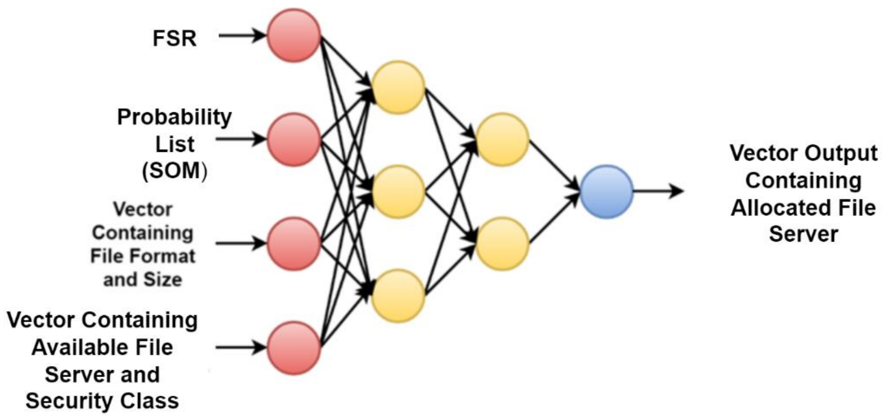


Figure 3.4 NN to Predict the Appropriate File Server to store a Multimedia File.

3.3 Experimental Evaluation

Internet was searched for a dataset that could be used to train all the DNNs of the proposed framework. There is no dataset available online with the attributes required in the framework for its experimental evaluation. So, a testbed was created for the experimental evaluation as displayed in Figure 3.5. The packages used in all the DNNs are common used packages like NumPy, pandas, matplotlib, and TensorFlow on python 3.7 framework coded in the Jupyter notebook [70]. Figure 3.5 also list the main packages used by a component for its accurate working.

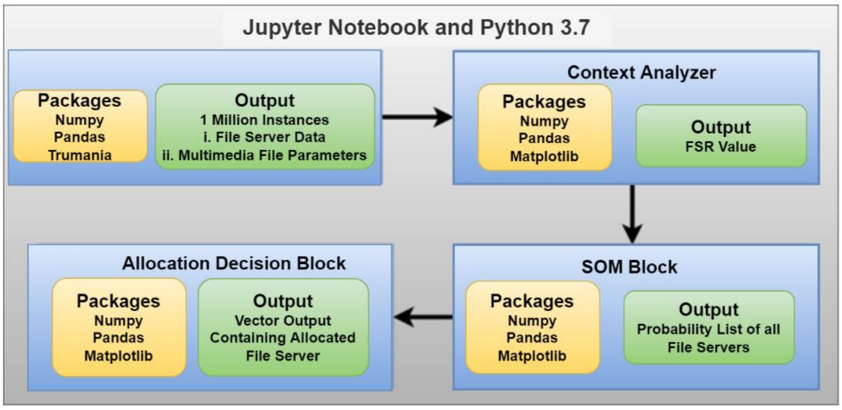


Figure 3.5 Testbed for Performance Evaluation of the Proposed Framework

However, the generation of synthetic dataset was one of the main challenges which was done using a package called Trumania. All file parameters are somehow related to one another, so a scenario based synthetic data generator was the best choice as compared to the schema based to generate the synthetic dataset. Trumania package uses the casual relationships to generate multiple permutations and combinations of various attributes to finally generate many instances in a dataset. Furthermore, this package automatically generates the relevancy among the attributes to find new relationships, so this was best suited for our synthetic dataset. This package outputs a file in the format of csv and its snapshot is displayed in Figure 3.6 and Figure 3.7. The dataset created contains all the required information for the training and testing of proposed DNNs, depicted in Figure 3.1. Ten datasets were created using the Trumania package and stored as separate csv files.

FILE_ID	WHEN	WHAT_IS_SEND	SENDING_LOCATION	SENDER	RECEIVER	SIZE	ESR
File_0000000000	1/06/2019 1:14	document	office	Ms. Michelle Curry	Dr. Jessica Stone	66.062432	Low
File_0000000001	1/06/2019 1:23	document	office	Melissa Escobar	Daniel Stanton	4.9887647	Medium
File_0000000002	1/06/2019 1:37	data_sheet	public_network	Marissa Smith	Jeffrey Jordan	44.853422	High
File_0000000003	1/06/2019 1:33	document	public_network	Grace Henderson	Mrs. Alexandra Green MD	152.15646	Low
File_0000000004	1/06/2019 1:08	document	office	Cameron Gregory	Dustin Harris	9.6190172	Low
File_0000000005	1/06/2019 1:13	data_sheet	mobile_network	Nancy Horton	Elizabeth Ryan	28.157246	High
File_0000000006	1/06/2019 1:35	video	office	Amber Hall	Vanessa Harvey	99.079876	High
File_0000000007	1/06/2019 1:24	data_sheet	public_network	Tracy Nguyen	Lori Munoz	27.369401	High
File_0000000008	1/06/2019 1:01	document	office	Kathleen Mcclure	Linda Dalton	117.83746	High
File_0000000009	1/06/2019 1:41	document	office	Lisa Key	Shelby Smith	75.57257	V_High
File_0000000010	1/06/2019 1:53	document	office	Anna Mendoza	Nicole Levy	37.612691	High
File_0000000011	1/06/2019 1:56	document	office	Daniel Green	Leslie Hodges	2.6793621	Medium
File_0000000012	1/06/2019 1:35	data_sheet	mobile_network	Carl Martinez	Valerie Peterson	31.507439	High
File_0000000013	1/06/2019 1:17	video	office	Bailey Valentine	Jessica Moody	115.54726	V_low
File_0000000014	1/06/2019 1:39	audio	office	Melanie Smith	Monique Hines	81.418494	Low
File_0000000015	1/06/2019 1:43	audio	home	Samuel Clark	John Scott	24.033623	Medium
File_0000000016	1/06/2019 1:32	document	mobile_network	Eric Martinez	Adam Jordan	39.909461	Low
File_0000000017	1/06/2019 1:09	document	office	Brianna Castaneda	Wendy Johnson	51.8359	High
File_0000000018	1/06/2019 1:48	document	office	Katherine Torres	William White	129.84636	V_High
File_0000000019	1/06/2019 1:40	document	office	Evan Peters	Daniel Warner DDS	47.497974	Low

Figure 3.6 Part of Synthetic Dataset Generated using Trumania

File_server_id	Time	File_server_location	Avail_memory	File_server_security
Fog_0000000000	01-06-2019 01:14	mobile_network	42.13344039	V_low
Fog_0000000001	01-06-2019 01:23	office	132.9355872	Low
Fog_0000000002	01-06-2019 01:37	public_network	33.07669352	V_High
Fog_0000000003	01-06-2019 01:33	office	6.464884017	High
Fog_0000000004	01-06-2019 01:08	home	11.74179817	Low
Fog_0000000005	01-06-2019 01:13	mobile_network	32.42746154	High
Fog_0000000006	01-06-2019 01:35	office	79.23194506	Low
Fog_0000000007	01-06-2019 01:24	home	61.44034835	Medium
Fog_0000000008	01-06-2019 01:01	office	120.9649694	Medium
Fog_0000000009	01-06-2019 01:41	home	162.7428136	Low

Figure 3.7 A Part of Dataset for File Servers

FILE_ID	WHEN	WHAT_IS_SEND	SENDING_LOCATION	SENDER	RECEIVER	SIZE	ESR	FSR
File_0000000000	1/06/2019 1:14	document	office	Ms. Michelle Curi	Dr. Jessica Stone	66.062432	Low	0.2
File_0000000001	1/06/2019 1:23	document	office	Melissa Escobar	Daniel Stanton	4.9887647	Medium	0.24
File_0000000002	1/06/2019 1:37	data_sheet	public_network	Marissa Smith	Jeffrey Jordan	44.853422	High	0.83
File_0000000003	1/06/2019 1:33	document	public_network	Grace Henderson	Mrs. Alexandra (	152.15646	Low	0.31
File_0000000004	1/06/2019 1:08	document	office	Cameron Gregory	Dustin Harris	9.6190172	Low	0.2
File_0000000005	1/06/2019 1:13	data_sheet	mobile_network	Nancy Horton	Elizabeth Ryan	28.157246	High	0.78
File_0000000006	1/06/2019 1:35	video	office	Amber Hall	Vanessa Harvey	99.079876	High	0.78
File_0000000007	1/06/2019 1:24	data_sheet	public_network	Tracy Nguyen	Lori Munoz	27.369401	High	0.84
File_0000000008	1/06/2019 1:01	document	office	Kathleen Mcclure	Linda Dalton	117.83746	High	0.68
File_0000000009	1/06/2019 1:41	document	office	Lisa Key	Shelby Smith	75.57257	V_High	0.72
File_0000000010	1/06/2019 1:53	document	office	Anna Mendoza	Nicole Levy	37.612691	High	0.7
File_0000000011	1/06/2019 1:56	document	office	Daniel Green	Leslie Hodges	2.6793621	Medium	0.24
File_0000000012	1/06/2019 1:35	data_sheet	mobile_network	Carl Martinez	Valerie Petersor	31.507439	High	0.89
File_0000000013	1/06/2019 1:17	video	office	Bailey Valentine	Jessica Moody	115.54726	V_low	0.12
File_0000000014	1/06/2019 1:39	audio	office	Melanie Smith	Monique Hines	81.418494	Low	0.24
File_0000000015	1/06/2019 1:43	audio	home	Samuel Clark	John Scott	24.033623	Medium	0.43

Figure 3.8 FSR Value is Calculated for all the Files

After the generation of the synthetic dataset the context analyser block was trained using the dataset and parameters listed in Table 3.2. Contextual values generated using Trumania package were encoded using one hot encoder. The output generated by this block, FSR was added as a new column as shown in Figure 3.8. All the ten-dataset created were used with the trained NN and it achieved an average accuracy of 81.5%, shown in Figure 3.9. After achieving the FSR value for all the instances, SOM lattice cluster was created and trained using the DNN. It also achieved the average accuracy of 83.4% as depicted in Figure 3.10. Lastly, the file parameters and intermediate values were used by the BPNN of allocation decision block which decides the file server where file could be stored. Allocation decision block was also trained for all the ten-datasets and achieved the average accuracy of 89.5% as shown in Figure 3.11. Figure 3.12 shows the result on the synthetic dataset with final allocation of storage servers to the multimedia files.

Table 3.2 DNNs Training Hyperparameters

Activation Function	Sigmoid
Optimization used	Adam Optimization
Output Units	1
Input Units	5
Momentum (β)	0.9
Hidden Layers	2
Learning rate (α)	0.5 initially
Mini-Batch Size	5000

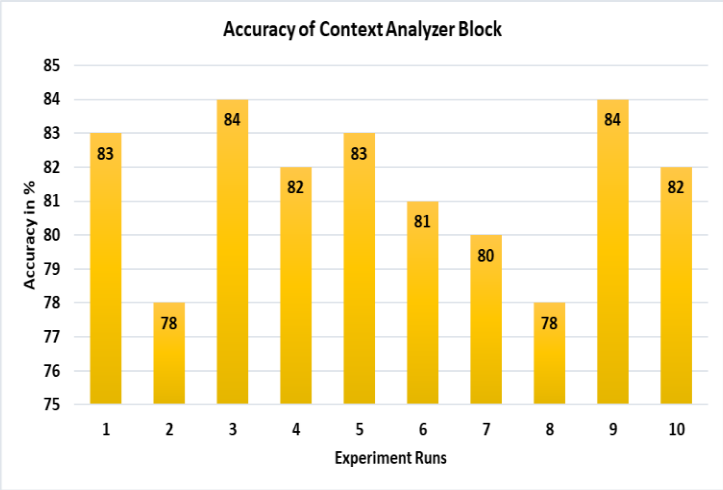


Figure 3.9 The Accuracy of Ten Runs for Context Analyzer Block

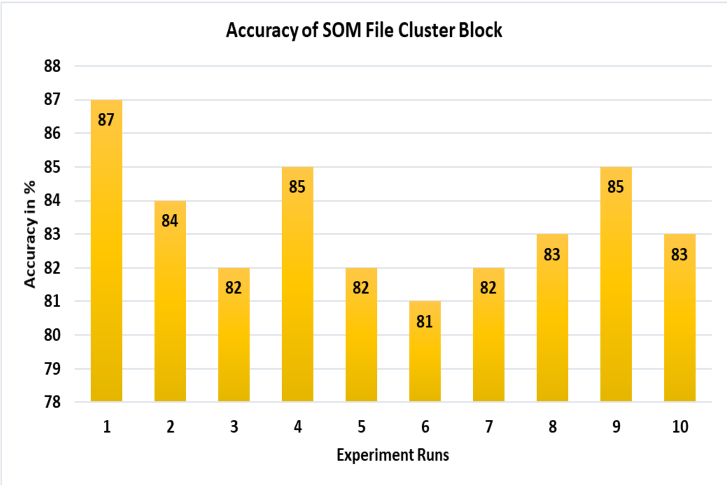


Figure 3.10 The Accuracy of Ten Runs for SOM Block

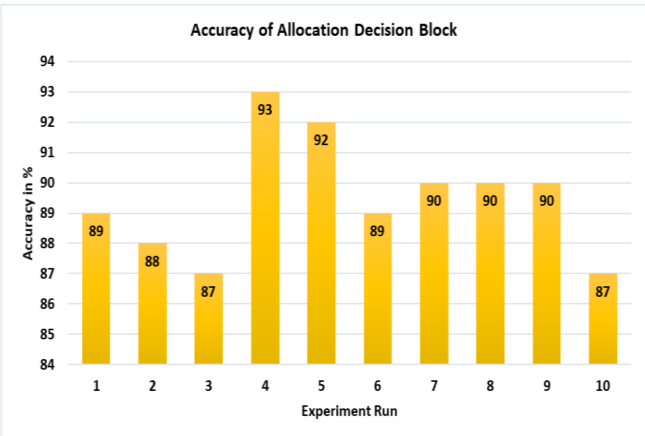


Figure 3.11 The Accuracy of Ten Runs for Allocation Block

FILE_ID	WHEN	WHAT_IS_SEND	SENDING_LOCATION	SENDER	RECEIVER	SIZE	ESR	FSR	Allocated File Server
File_0000000000	01-06-2019 01:14	document	office	Ms. Michelle Cur	Dr. Jessica Ston	66.06243	Low	0.2	File_server_0000000001
File_0000000001	01-06-2019 01:23	document	office	Melissa Escobar	Daniel Stanton	4.988765	Medium	0.24	File_server_0000000008
File_0000000002	01-06-2019 01:37	data_sheet	public_network	Marissa Smith	Jeffrey Jordan	44.85342	High	0.83	File_server_0000000003
File_0000000003	01-06-2019 01:33	document	public_network	Grace Hendersor	Mrs. Alexandra	152.1565	Low	0.31	File_server_0000000004
File_0000000004	01-06-2019 01:08	document	office	Cameron Gregor	Dustin Harris	9.619017	Low	0.2	File_server_0000000019
File_0000000005	01-06-2019 01:13	data_sheet	mobile_network	Nancy Horton	Elizabeth Ryan	28.15725	High	0.78	File_server_0000000005
File_0000000006	01-06-2019 01:35	video	office	Amber Hall	Vanessa Harvey	99.07988	High	0.78	File_server_0000000015
File_0000000007	01-06-2019 01:24	data_sheet	public_network	Tracy Nguyen	Lori Munoz	27.3694	High	0.84	File_server_0000000010
File_0000000008	01-06-2019 01:01	document	office	Kathleen Mcclure	Linda Dalton	117.8375	High	0.68	File_server_0000000018
File_0000000009	01-06-2019 01:41	document	office	Lisa Key	Shelby Smith	75.57257	V_High	0.72	File_server_0000000022
File_0000000010	01-06-2019 01:53	document	office	Anna Mendoza	Nicole Levy	37.61269	High	0.7	File_server_0000000026

Figure 3.12 File Servers Allocated to Every File

3.4 Conclusion

Methods and framework are needed which can convert high dimensional raw contextual data into low dimension so that it can be better utilized by the recommender systems for the generation of effective recommendations. In this chapter, DNNs based framework was designed which uses the raw context data and frame single value known as FSR which was further utilized by the SOM cluster and probabilities of allocation to certain file servers were generated. All this processed context values were effective in finding the best file server for any file placement depending on its security and privacy requirements.

CHAPTER 4

ANALYSIS OF CONTEXTUAL AND NON-CONTEXTUAL DATA USING DNNs

As discussed in the introduction section, traditional RS need to perform context engineering of the raw contextual data before using it in the recommendation generation phase. Contextual engineering consists of many steps which can be context pre-processing, contextual feature extraction and formation, conversion of context in a specified format and many others. These steps depend on the type and source of the collection of the contextual data. With the proliferation of digital sensors and smart things, contextual data for any CARS have heterogenous formats and multiple contradicting contextual data points. Furthermore, the traditional RS using traditional feature extraction methods may skip some of the features which are rarely used by any CARS. These features may not be important for most of the CARS but for some CARS they may be crucial.

From the above points, it may be concluded that frameworks are required which can perform context engineering and recommendation generation as a single entity. These frameworks would input all contextual and non-contextual data points at the same time and learn the patterns specific to the individual user. These frameworks would also increase the adoption of CARS with more accurate and precise recommendation generations. Furthermore, these frameworks will not require any separate context engineering, so, their convergence and inference time will be less as compared to the traditional RS.

DNNs can be used for the above said frameworks because they can learn complex patterns from large amount of data [71]. Also, the contextual and non-contextual data can be input to the DNNs so that they can be trained for individual user. DNNs have different frameworks for different type of input and output required from a CARS. For example, sequential patterns can be learned using Recurrent Neural Network (RNN) and images can be better trained on CNN frameworks. DNNs can also be used to generate intermediate outputs which can be used by another DNN in the same framework depending on the type of contextual and non-contextual data points.

In this chapter, three frameworks are proposed which uses contextual and non-contextual data parallelly to generate recommendations using DNNs. First framework generates the recommendation of the water requirement of an agriculture field. The second framework recommends the best posture for a user and generate alert if user is in wrong posture using

DNNs. Lastly, a framework is proposed for smart epidemic control which generates intermediate outputs using different DNN frameworks based on the type of input data. All the proposed frameworks are tested for their respective datasets, and they proved the hypothesis that contextual and non-contextual data can be used in DNNs at the same time to generate effective recommendations.

4.1 Smart Agriculture using CARS

The main goal of any government in the world is to provide necessities to its citizens. Food is among one of the basic requirements for a human being [72]. This requirement has drawn attention towards agriculture sector and to increase the crop productivity [73] [74]. The crop productivity depends upon various data elements such as evaporation rate, sun duration, soil quality, wind speed, soil moisture, humidity, cloud covers, rainfall, and temperature. These data elements affect various crop factors such as irrigation requirements, soil suitability, optimum quality of fertilizers and pesticides, appropriate rowing, and maturity date of a crop [75]. In current era of communication and technology the meteorological data elements can be easily collected using multiple sensors. Many crop yielding parameters like soil suitability, maturity date, rowing time, quantity of fertilizers and irrigation requirements depends on the effective analysis of these environmental and meteorological data elements [76]. A context sensitive analysis of these factors can result in the improvement of various crop yielding parameters and could increase crop productivity [77]. Furthermore, these can also help to reduce the soil, air and other kind of pollutions which are caused by excessive usage of fertilizers and pesticides.

So, an effective context-sensitive framework is required for prediction of various crop factors that can generate recommendations based on the current context i.e., by analysing current metrological and environmental factors [78]. However, in agriculture the context is an important factor because the trained NN for USA cannot work for India due to difference in environment, meteorological factors, techniques, and methods of farming. Past analysis and recommendation should also be considered in the generation of new recommendations apart from the current sensor values.

To achieve the above said objective a LSTM based NN is designed in this chapter which uses various environmental and meteorological contextual and non-contextual data points and predicts the irrigation requirement for any crop. The LSTM is used in the proposed framework because the collected data points are of time series data and past input has considerable influence on the prediction of future irrigation requirement.

4.1.1 The Agriculture CARS Framework

The three major components of the proposed framework are shown in Figure 4.1 which are collection of input data, prediction using LSTM and the final prediction representation. The input for the proposed framework is from environmental and meteorological department websites and various sensor APIs are used to collected data from other sources like google weather etc. Further, meteorological data for a given location is recorded from the government websites. These input data contain both the contextual and non-contextual data points. The DNN designed in the proposed framework takes all the inputs from the input component and perform feature engineering to decide the important input factors for the model which is then fed as input to the LSTM model for generation of final recommendations.

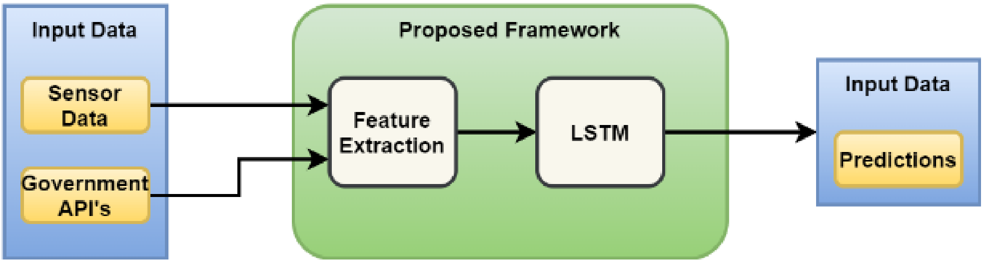


Figure 4.1 Context Sensitive Framework for Prediction of Irrigation Requirement

4.1.1.1 Contextual and Non-Contextual Input Data

A crop can be affected by various parameters which may be contextual or non-contextual as listed in the Table 4.1. There are more than sixteen parameters with variable values. Many of these parameters depends on one another, for example the maximum temperature has direct relationship with the evaporation rate. Therefore, it is better to reduce these parameters into a few parameters which are relevant for a given crop.

4.1.1.2 Dimension Reduction

The Table 4.1 has listed the sixteen different factors which may affect any crop, however there can be more parameters. Furthermore, the effect of any contextual or non-contextual parameter on the irrigation requirement of any crop is not same for all the crops. Some crops may be affected from the maximum temperature and other may have more impact of the minimum temperature. It is better to extract the features which affects any crop in larger context and should focus on analysis of these features only. Furthermore, as LSTM model considers large number of previous inputs while making prediction, it would be complex hyperparameter tuning and training with large number of input features. In the proposed framework, a DNN is used for the extraction of important features as it was done in Chapter 3. A BPNN was designed with all the sixteen inputs, and it outputs an encoder with five values which will be further use by the LSTM network for recommendation generations.

Table 4.1 Inputs Affecting any Crop

Meteorological	**Contextual/ Non-contextual**	**Meteorological**	**Contextual/ Non-contextual**
Latitude of the location	Non-contextual	Condensation rate	Contextual
Altitude of the location	Non-contextual	Evaporation rate	Contextual
Cloud Cover at any point	Contextual	Trend of the rainfall	Contextual
Distance from sea	Non-contextual	Crop type	Non-contextual
Speed of the wind	Contextual	Level of the moisture	Contextual
Duration of the sunlight	Contextual	Soil colour	Non-contextual
Amount of ocean current	Contextual	Context of the Organic Matter	Contextual
Soil PH	Contextual	Land Slope	Non-contextual

4.1.1.3 LSTM based Prediction

LSTM are better suited for time series data analysis because they can consider the output at timestamp t_0 for the input of the model at time t_1 [79], [80]. The environmental and meterological data which will be collected in the proposed framework will be a time series data where most of these features has impact of the previous values when any prediction need to be made for the future. The overall concept of analysis of time series data using LSTM can be seen from the Figure 4.2. The hidden layer h_1 takes all the inputs from timestamp t_1 and has an input from the hidden layer of timestamp t_0. This makes LSTM better suitable for analysis of time series contextual data.

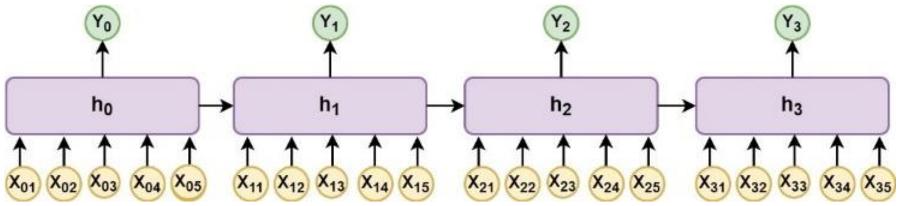


Figure 4.2 Connection of Various Blocks of Different Timestamps for LSTM

4.1.2 Experimental Evaluation and Performance Results

Wheat and sunflower crop in Haryana and Punjab state of India are considered for the experimental evaluation and performance result for the recommendation of the irrigation requirement using the data collected from Google weather API's and data available on the Haryana and Punjab state websites. As the explanation of the proposed framework, the experimental section is also divided into three subcomponents which are contextual and non-contextual data collection, feature extraction, and prediction.

Figure 4.3 (a) Pehowa (b) Gurdaspur Geographical Location for which Data was Collected

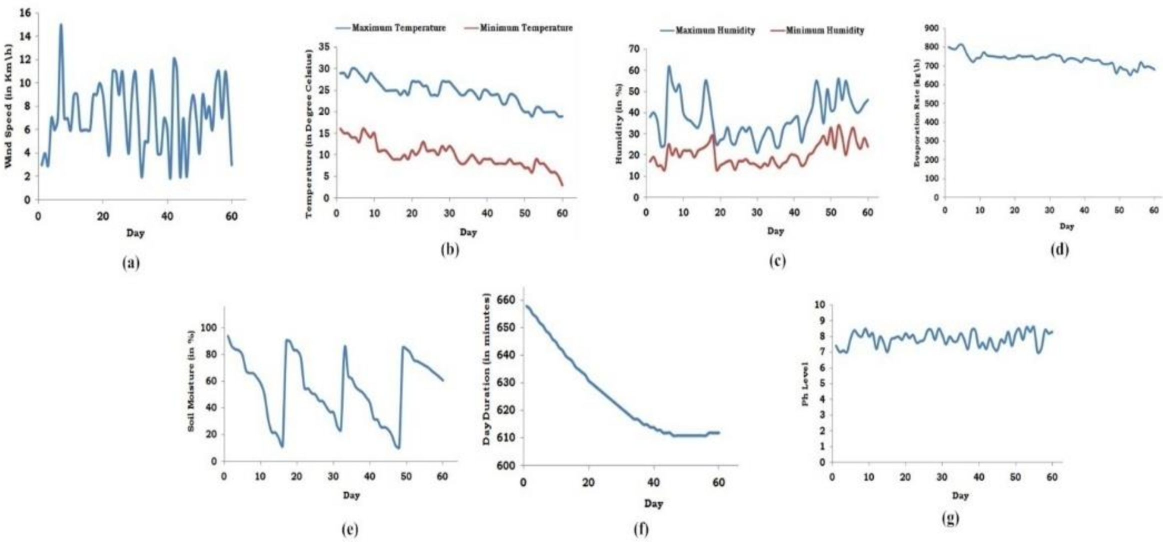


Figure 4.4 Meteorological Data Representation of Sixty Days

4.1.2.1 Contextual and Non-Contextual data collection

Data is collected from two major northern states of Indian which are Haryana and Punjab. The exact parameters of the selected two-location are listed in Table 4.2 and Figure 4.3. The output of the proposed framework from the experimental evaluation is the recommendation of the irrigation requirement into one of the classes of high, mid and low. The ninety days data is collected from the Google API and the government websites. A train test split of 70/30 was used to train and test the proposed DNN Figure 4.4 shows different types of data collected from the sources.

Table 4.2 Different Parameters of Two Selected Cities

	Cities	
	Pehowa	Gurdaspur
Land Slope	Level	Level
Crop Type	Wheat	Sunflower
Sea Distance	1050 KMs	1270 KMs
Soil Color	Pale Brown	Dark Brown
Location Altitude	310	223
Location Latitude	28° 12' N	32° 03' N

4.1.2.2 Feature Extraction

A numeric value is calculated for all the contextual and non-contextual features, top five high ranked features are shown in the Table 4.3. The selected features in Table 4.3 are used by the LSTM for making irrigation recommendation for wheat and sunflower crops.

Table 4.3 Dimension Reduction of Contextual and Non-contextual Data

Wheat in Pehowa, India		Sunflower in Gurdaspur, India	
Rank Value	Attribute	Rank Value	Attributes
0.114	Lowest Temperature	0.134	Lowest Temperature
0.187	GDD	0.201	Rate of Evaporation
0.224	Evaporation Rate	0.221	GDD
0.341	Maximum Temperature	0.287	Highest temperature
0.625	Soil Moisture	0.667	Soil Moisture

4.1.2.3 LSTM based Prediction

Table 4.4 shows the parameter and hyperparameters used for the training of the LSTM network for the recommendation generation. Figure 4.5 shows the error rate and accuracy of the LSTM model while training where the model achieved the accuracy of around 85% after the 500 epochs of training. While the normal BPNN can achieve only 72% accuracy after same number of epochs as shown in Figure 4.6.

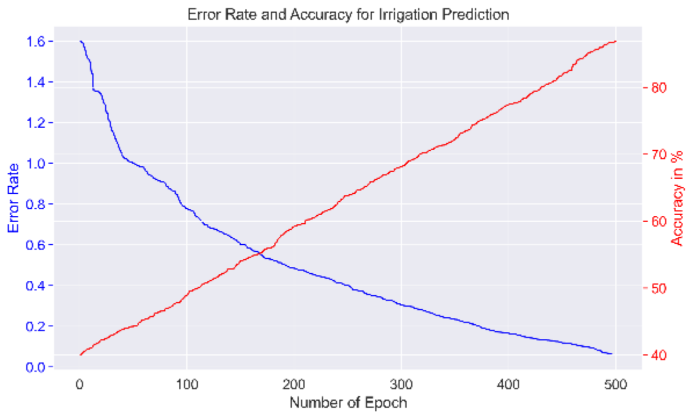


Figure 4.5 Accuracy and Error Rate for Recommendation Generations of LSTM

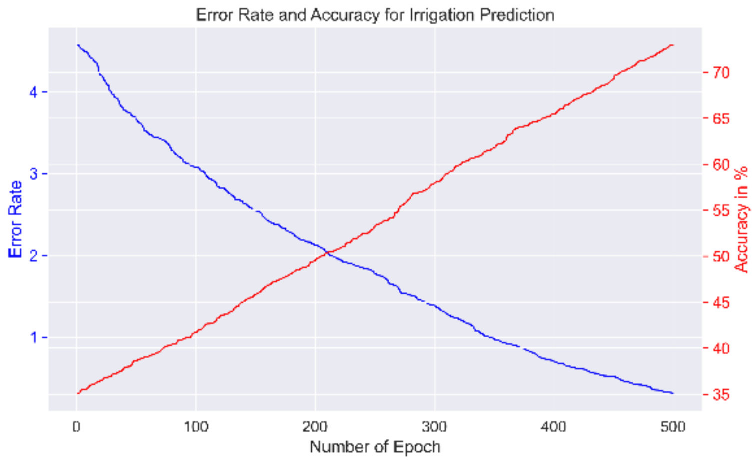


Figure 4.6 Accuracy and Error Rate for Recommendation Generations of BPNN

4.2 Sleep Posture Prediction

Another framework is proposed which takes both contextual and non-contextual inputs parallelly and generate recommendations. The data used in this proposed framework is pre-processed data and does not require any feature engineering or dimension reduction. This is done so that the main focus is on designing effective framework for analysis of contextual and non-contextual data, however, if required context engineering can be easily done using DNNs as shown in previous sections and chapters.

Sleep is must for everyone, and a quality sleep is needed for health body and mind. Sleeping posture has a vital role in getting quality sleep [81] [82]. A bad posture which may be due to wrong habit or mattresses may lead to origination of a lot of diseases [83] [84]. Everyone knows the importance of sleep; but the continuous monitoring of a user's posture is not feasible as it is a private thing and the amount of manpower required is not possible to obtain [85]. Furthermore, a user can change his/her posture at any time during the sleep. So, a CARS framework is required which can generate alerts when any user is not sleeping in correct posture.

A common sleep posture cannot be generated for all the users, because everyone has different posture where they can have sound sleep. Further, a doctor can advise different posture to different users based on their other medical histories and requirements. This generates the need of reading and analysing the contextual information of a user for posture detection and alert generation process. Fortunately, many hardware [86] and sensors are available in the marketplace which can record different values [87] during the sleep of a user. However, a framework is required which can generate effective context sensitive recommendations. Also, the comfort level of the user should be calculated in real-time using various body sensors [88] [89]. The detection of the posture is also a latency sensitive task, for an example the detection of wrong posture after thirty minutes or more is of no use to the user and may be harmful for the health of the user [90][91].

To achieve the objective of latency sensitive analysis of contextual and non-contextual data, a framework is proposed in this chapter which uses cloud for the training of the user specific NN. Fog computing [92] environment hosts the trained NN which is used to generate alerts in timely fashion [93]. DNNs are used in the proposed framework for the analysis and recommendation generations from the collected data points.

4.2.1 Sleep Posture CARS Framework

The various components designed in the proposed framework are the sensors collecting data, cloud computing for training of DNNs and fog computing for the generation of real time alerts, shown in Figure 4.7.

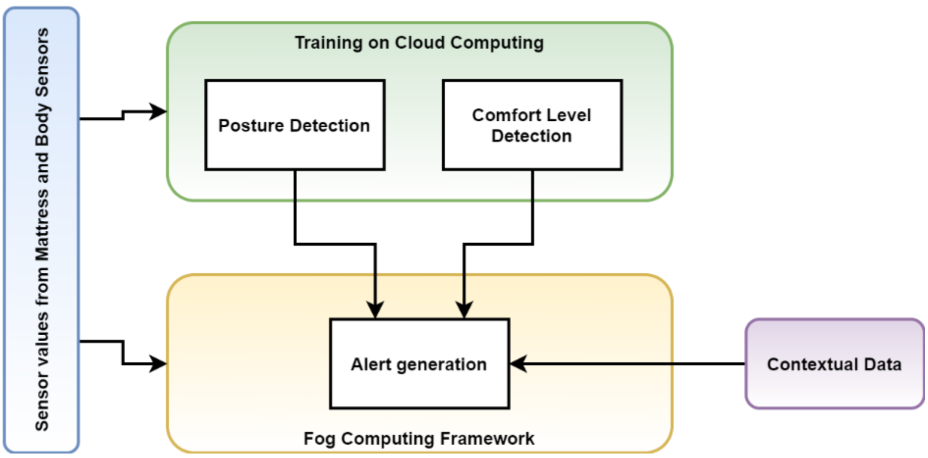


Figure 4.7 Framework to Detect and Alert Wrong Posture

4.2.1.1 Sensor Input Data

A complete and concise dataset was not available on the internet which can be directly used in the proposed framework. However, a hybrid dataset was created by joining two different datasets where first dataset contains mattresses datapoints using which posture of the user [87], [94] can be identified and the second dataset is the sleep dataset which contains the sleep comfort data points based on the body sensors [95] [96].

4.2.1.2 Posture Detection

The detection of the posture in a night sleep is a temporal data which is varied based on the time spent in the sleep. This makes the posture data a sequential data where the sequence of posture in which user is sleeping is important for the generation of alerts. Therefore, the LSTM is used in the proposed framework for the detection of posture of any user. RNN could also be used for the DNN model, but the experimental evaluation gave better results with the LSTM. The reason for better LSTM output is because it can store large number of previous values and does not suffer from vanishing gradient problem as it was with RNN model [97].

4.2.1.3 Sleep Comfort Prediction

The prediction of sleep comfort is a simple task as compared to the posture detection, so a BPNN is used in this section. CAP sleep dataset [96] contains various health parameters related to comfort of sleep along with proper labelling. So, this dataset is used in this component.

4.2.1.4 Contextual Data

Every user has different posture in which he/she have a comfortable sleep and a doctor may have suggested a specific postures for a user based on their health requirements. So, collection of current contexts of the user is important for the successful working of CARS proposed in this chapter. Based on the 17 different postures given in [94] a hot encoder is created which has value of 1 for the postures which are allowed for the user and 0 for the others. Furthermore, the allowed time in every posture is also recorded in the hot encoder. These are very important for the user specific alert generations.

4.2.2 Performance Evaluation

A testbed, as shown in Figure 4.8, has been proposed to experimentally evaluate the proposed framework, the two DNN models used are deployed on local cloud and public Azure cloud using Aneka PaaS software [98]. DNNs created using Tensorflow and Juypter notebook are trained using the training data on the Microsoft Azure platform and then the trained models are deployed on local machines for the generation of context aware user specific recommendations for posture correction and alert generations. Transfer learning methodology helps to train DNN models better on the public cloud and then use them directly on the fog layer [92] for the prediction process.

4.2.2.1 Data Collection

Vista medical FSA Softflex mattresses containing 2048 sensor points is used to collect two-minute data with 120 frames of all the sensor points. The age bracket of users for this data collection is from nineteen years to thirty-four years. On the other dataset, one hundred and eight polysomnographic recordings have been recorded for various health parameters such as EMG, EKG, respiration signals, and 2 EEG channels. The collected dataset is linked to one user and divided into training and testing datasets using 70/30 ratio. Furthermore, the user contextual data is generated using the random generators and attached to the already created datasets.

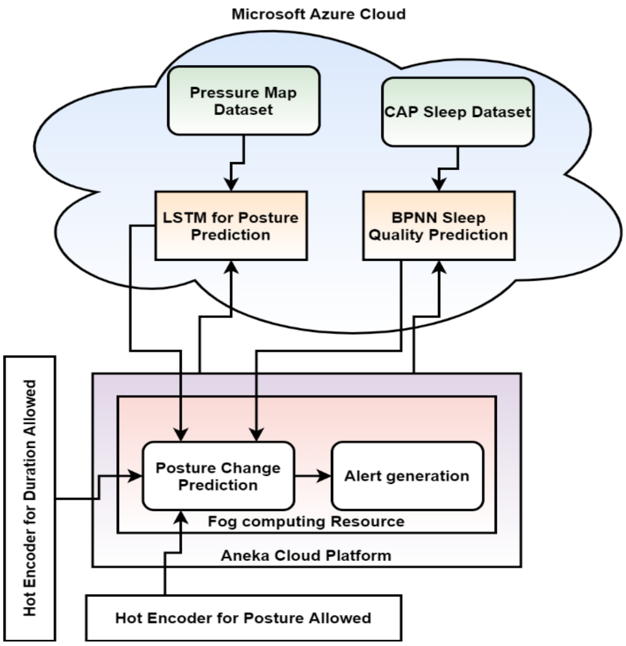


Figure 4.8 The Proposed Framework Experimental Testbed

4.2.2.2 Training Parameters

A DNN has various parameters and hyperparameters, Table 4.4 shows the parameters and hyperparameters used in the proposed framework for the training of the DNNs on the public cloud.

Table 4.4 Parameter and Hyperparameters of the DNNs

Parameter/Hyperparameters	Posture Detection	Sleep Quality
Type of DNN	LSTM	BP NN
Type of Tensorflow Model	Sequential	Sequential
Size of Input	32*64	9
Size of Mini-Batch	None	None
Activation Function of Output	Leaky ReLu	Sigmoid
Activation Function of Hidden Layers	ReLu	ReLu
Loss Function	MAE	MAE
Optimization Used	Short-Adam	Short-Adam
Hidden Layers	3	1
Metric Used	Accuracy	Accuracy
Size of Output	13	3

Table 4.5 Accuracy for All Datasets in the Proposed DNN

	Accuracy with datasets of		
	Training Dataset	Development Dataset	Validation Dataset
Posture DNN Training	89%	83%	
Comfort DNN Training	87%	84%	
Posture Detection			90%
Alert Generation DNN			95%

4.2.2.3 Experimental Results

Due to non-existence of the class imbalance problem in the dataset, accuracy can be effectively used to find and compare the output of the proposed DNNs. The dataset discussed in previous sections is increased using probability based bootstrapping techniques so that training-testing split is adequate for the experimental evaluation. The dataset after bootstrapped is divided into three categories which are validation, development, and testing datasets. The training of the comfort and posture prediction NNs is conducted using the training dataset which achieved the training accuracy of 87% and 89% respectively. A cross validation of ten was performed with the development dataset for the evaluation of the trained NNs which also achieved the average accuracy of 84% and 83%. To validate the trained and tested models, they were deployed on the fog layer of the proposed testbed and validation dataset was used to make predictions. The NN at the fog layer got the 90% and 95% accuracy for the posture detection and the alert generation respectively. Table 4.5 shows the respective results.

4.3 Smart Epidemic Control

The previous two sections discuss simpler DNN models for analysing contextual and non-contextual information which resulted in high level of accuracy. In this section, a DNN framework is designed which has sequences of DNNs to analyse various contextual and non-contextual information [99][100]. In the proposed framework, a use case of smart epidemic control is discussed in which various inputs are used to predict the colour marking of a geographical area. Many properties of the spread of various infectious diseases are common such as direction of spread, intensity of infection, R-value etc [101]–[109]. So, a common framework can be easily designed which can cover most of the infectious spreads [110], [111]. The designed framework is for any kind of epidemic, however, for proof of concept a use case of Zika virus spread in Brazil is considered [112]. The proposed framework can be part of a large Smart Epidemic Control (SEC) system which may generate various recommendations. The role of context in any SEC is very important for its successful implementation and working. A non-contextual SEC will not be able to generate real time recommendations and hence will not be worthy to be implemented. As in the previous chapters and sections, various types of DNNs are used in the proposed frameworks depending on the output requirement and input formats to generate the effective recommendations.

In the proposed framework, the geographical area is classified into smaller regions using the hexagonal structures. Firstly, the spread of infectious is classified into outbreak, pandemic and epidemic, etc. based on various inputs like density of infection, population size, number of cases etc. Parallelly, it takes the input of number of infections in current hexagon and its neighbour hexagons to recommend the spread direction. Final NN takes all these inputs with the mode of spread of the disease and predicts the colour marking of the current hexagon. The number of colour marks used for hexagon classification can be of any amount, however in this chapter any hexagon is classified into three colour markings which are red, green, and yellow. This type of marking will guide the local authorities and the governments to focus on important areas. Furthermore, the areas will not be pushed into complete lockdowns, thus helping the economy.

4.3.1 SEC CARS Framework

Figure 4.9 shows the proposed framework with all its sub-components which takes both contextual and non-contextual information from the government authorities and mark any geographical areas. Hexagon structures are used to divide the geographical area into smaller regions, the structure is displayed in Figure 4.10 because it would be computational impossible to develop a CARS which can accurately colour mark large states and, also, it would not serve the purpose of focusing at small areas. Dividing any physical area into hexagons and deploying the proposed CARS in every hexagon is a better and feasible solution.

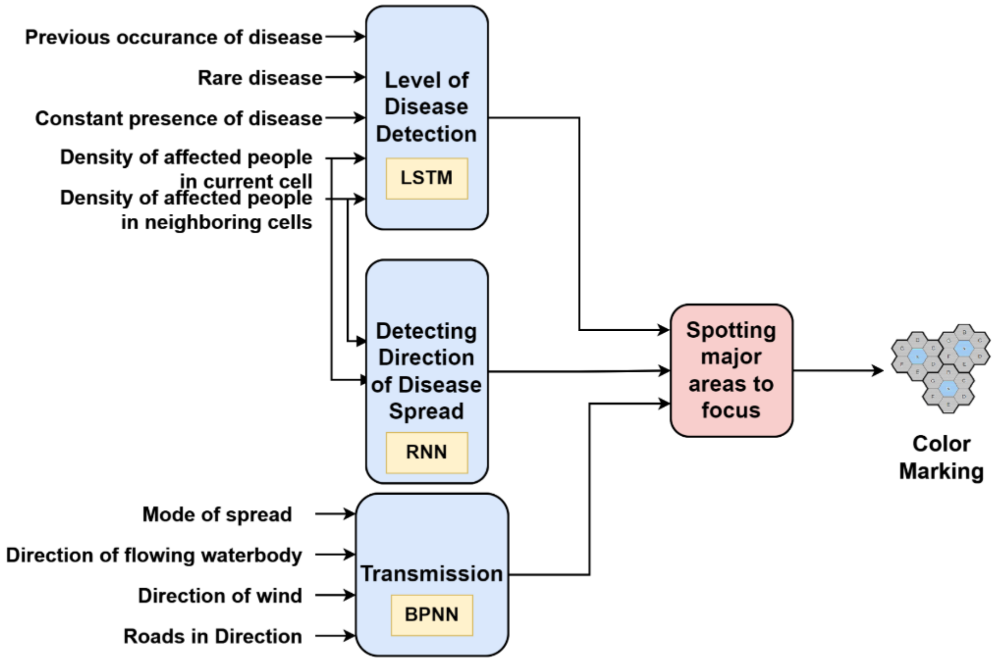


Figure 4.9 Context Aware Smart Epidemic Control System

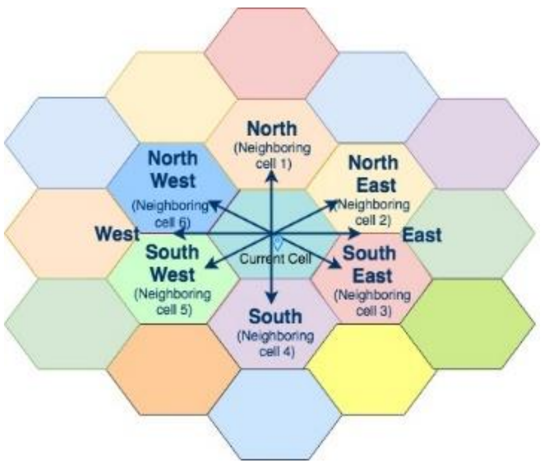


Figure 4.10 Area Division using Hexagonal Structure

4.3.1.1 Infection Level

In the proposed framework, seven types of infectious spreads are considered which are endemic, epidemic, outbreak, pandemic, sporadic, hyperendemic and cluster [113]. A LSTM [114] based NN is trained which takes inputs listed in Table 4.6 and Table 4.7 and assigns a category to every hexagon. Figure 4.11 shows different input to the network and output the categories.

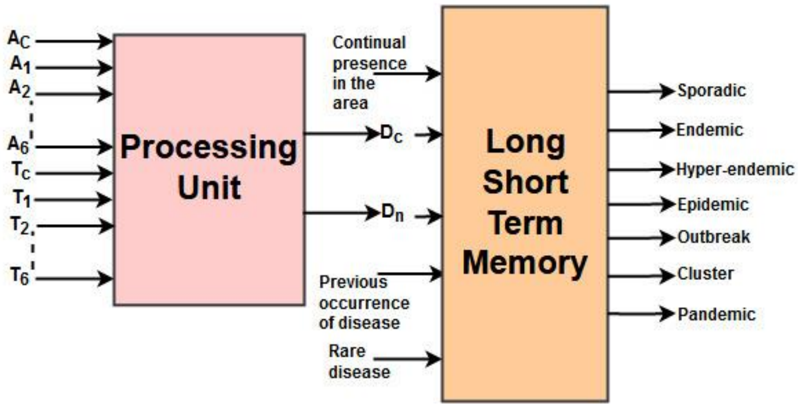


Figure 4.11 DNN for Infection level Prediction

DNN used for the infection level prediction has two parts which are the pre-processing and LSTM network. The pre-processing unit reads the value of number of infections in current hexagon and its neighbouring hexagons. Also, it reads the total population in all the neighbouring hexagons and the current hexagons. The pre-processing unit calculates the density of infection in the current hexagon and in its neighbours using the following equations:

$$\text{Density}_c = \text{Number of Infection}_c \,/\, \text{Total Population}_c \tag{4.1}$$

$$\text{Neighbor Density } (D_n) = \{(\text{Desnity}_1 + \text{Density}_2 + \ldots..+ \text{Density}_6)\ /6\} \tag{4.2}$$

All the notations used in the Figure 4.11 and in the above equations are explained in Table 4.6.

Table 4.6 Data Values Requiring Pre-processing

Input Symbol	Explanation
D_n	Neighboring hexagon density of spread.
T_C	Total population in current hexagon.
$T_1 \ldots.. T_6$	Neighboring hexagons population.
$A_1 \ldots. A_6$	Number of people infected in neighboring hexagons
D_C	Current cell infection density
A_C	Number of people infected in current hexagon

The output generated by the pre-processing unit is then used by the LSTM for recommending the infection level of the current hexagon. All the inputs and their respective notations are explained in Table 4.7. The LSTM is used in this component because the long sequences of past infections also needs to be considered for effective recommendation generation, which can only be done by using LSTM framework [114], [115].

Table 4.7 LSTM Inputs

Input	Explanation
Rare disease	A Boolean value stating whether the current spread is of a rare infection or not.
D_n	Neighboring hexagon density of spread
D_c	Current hexagon infection density
Last occurrence of disease	When this disease was last spread.
Constant presence of disease	Whether this infection is common in this area or not

4.3.1.2 Infection Direction

It would be easier for the government agencies to stop the infection spread and better control it, if they know the direction in which infection can spread. In the proposed framework, RNN model is used to predict the direction of spread of the infection based on the current and past values in a hexagon and its neighbours [116]–[118]. Figure 4.12 shows the RNN designed to predict the spread of infection from all the six possible directions.

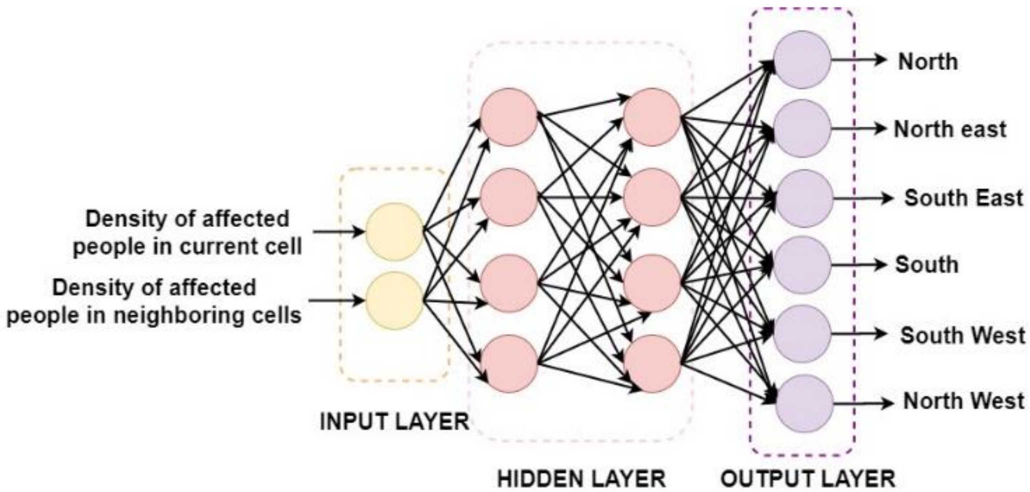


Figure 4.12 Structure of the DNN for Direction of Infection Prediction

4.3.1.3 Identifying mode of Transmission

The infectious diseases may have different or more than one mode of transmission from one human to another. Many infectious diseases spread using animals which are mostly found in water bodies, or a disease can spread through bacteria and other small particles in the air [100], [119]. So, in the proposed framework, a BPNN is designed which inputs mode of transmission in the form of a hot encoder and then takes input about various sources of spread like presence of water bodies, direction of airflow etc. This DNN provides recommendation that a certain infectious disease can be spread in the current hexagon or neighbouring hexagons apart from normal human contact. Inputs to this DNN are shown in Figure 4.9.

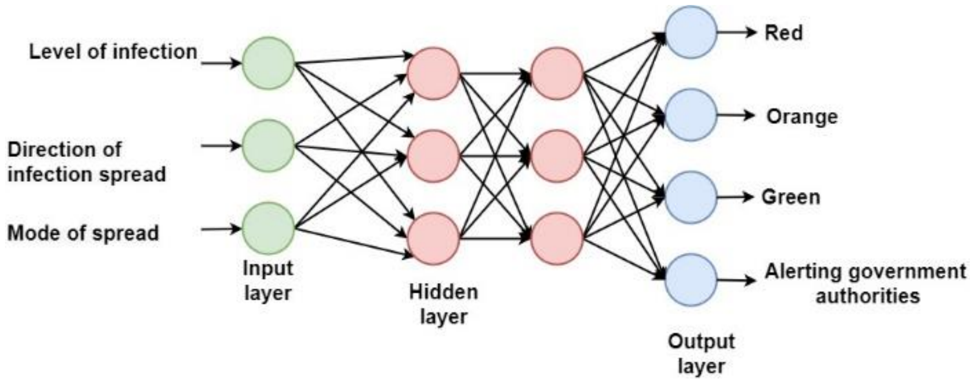


Figure 4.13 Final NN for Color Marking

4.3.1.3 Marking the hexagon

The final recommendation of the proposed framework is to assign a respective colour to every hexagon based on all the contextual and non-contextual data points. Three inputs are fed to the final NN as shown in Figure 4.13 which were generated from the other DNNs. This DNN recommends whether a given hexagon should be marked red, orange, or green. Where red represents the highest level of infection and green with the lowest level of infection and orange represents the moderate infection.

4.3.2 Experimental Setup

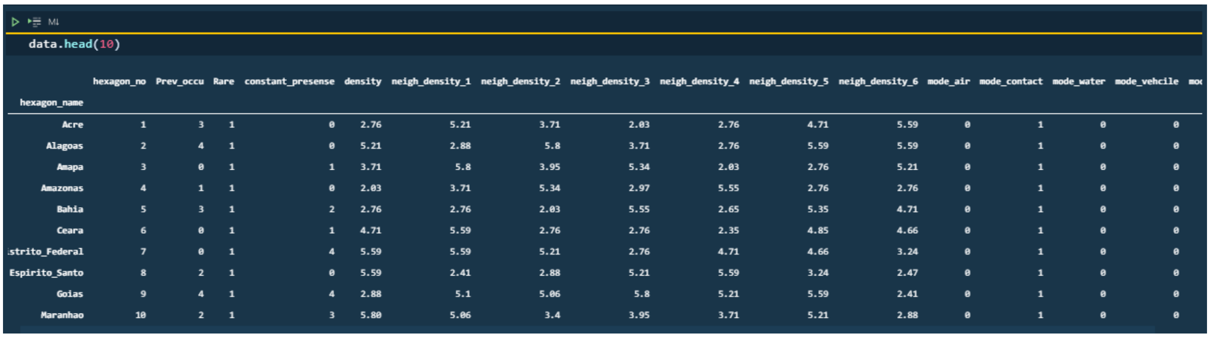
The proposed framework has multiple inputs of contextual and non-contextual data which was not available in a single dataset online. However, a dataset was identified which was collected during the Zika virus spread in Brazil. This section explains the creation of hybrid dataset, training, and testing of DNNs.

4.3.2.1 Dataset

For the experimental setup, a synthetic dataset was required which contains all the data inputs as required by the proposed framework. The data society has collected a Zika virus [112] spread dataset for countries like Brazil, Columbia and others which is categorized in region, district and provinces of these countries.

Table 4.8 Explanation of the Attributes of Experimental Evaluation Dataset

Attribute Name	**Type of the Value**	**Explanation**
Direction	Hot-Encoder	Prediction direction in which infection can spread.
Neighbor Density	Hot-Encoder	Neighboring hexagon density of spread
Density	Numeric Value	Current hexagon infection density
Constant Presence	Numeric Value	The number of times this infection spread in this hexagon in the last ten years.
Rare disease	Boolean Value	whether the current spread is of a rare infection or not.
Mode of Transmission	Hot-Encoder	Provides list of modes using which this infection can spread
Previous Occurrence	Numeric Value	When this disease was last spread.



```
data.head(10)
```

hexagon_name	hexagon_no	Prev_occu	Rare	constant_presense	density	neigh_density_1	neigh_density_2	neigh_density_3	neigh_density_4	neigh_density_5	neigh_density_6	mode_air	mode_contact	mode_water	mode_vehcile
Acre	1	3	1	0	2.76	5.21	3.71	2.03	2.76	4.71	5.59	0	1	0	0
Alagoas	2	4	1	0	5.21	2.88	5.8	3.71	2.76	5.59	5.59	0	1	0	0
Amapa	3	0	1	1	3.71	5.8	3.95	5.34	2.03	2.76	5.21	0	1	0	0
Amazonas	4	1	1	0	2.03	3.71	5.34	2.97	5.55	2.76	2.76	0	1	0	0
Bahia	5	3	1	2	2.76	2.76	2.03	5.55	2.65	5.35	4.71	0	1	0	0
Ceara	6	0	1	1	4.71	5.59	2.76	2.76	2.35	4.85	4.66	0	1	0	0
istrito_Federal	7	0	1	4	5.59	5.59	5.21	2.76	4.71	4.66	3.24	0	1	0	0
Espirito_Santo	8	2	1	0	5.59	2.41	2.88	5.21	5.59	3.24	2.47	0	1	0	0
Goias	9	4	1	4	2.88	5.1	5.06	5.8	5.21	5.59	2.41	0	1	0	0
Maranhao	10	2	1	3	5.80	5.06	3.4	3.95	3.71	5.21	2.88	0	1	0	0

Figure 4.14 Final Hybrid Dataset

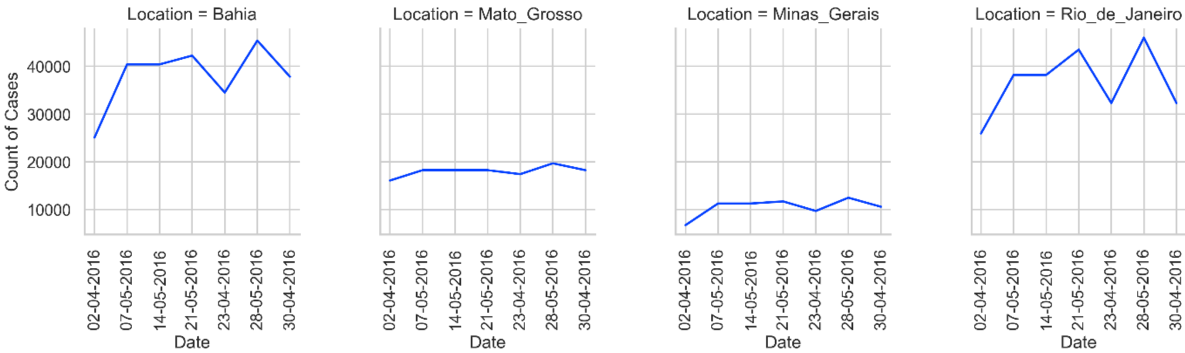


Figure 4.15 Infection Number where Infection was Largely Concentrated

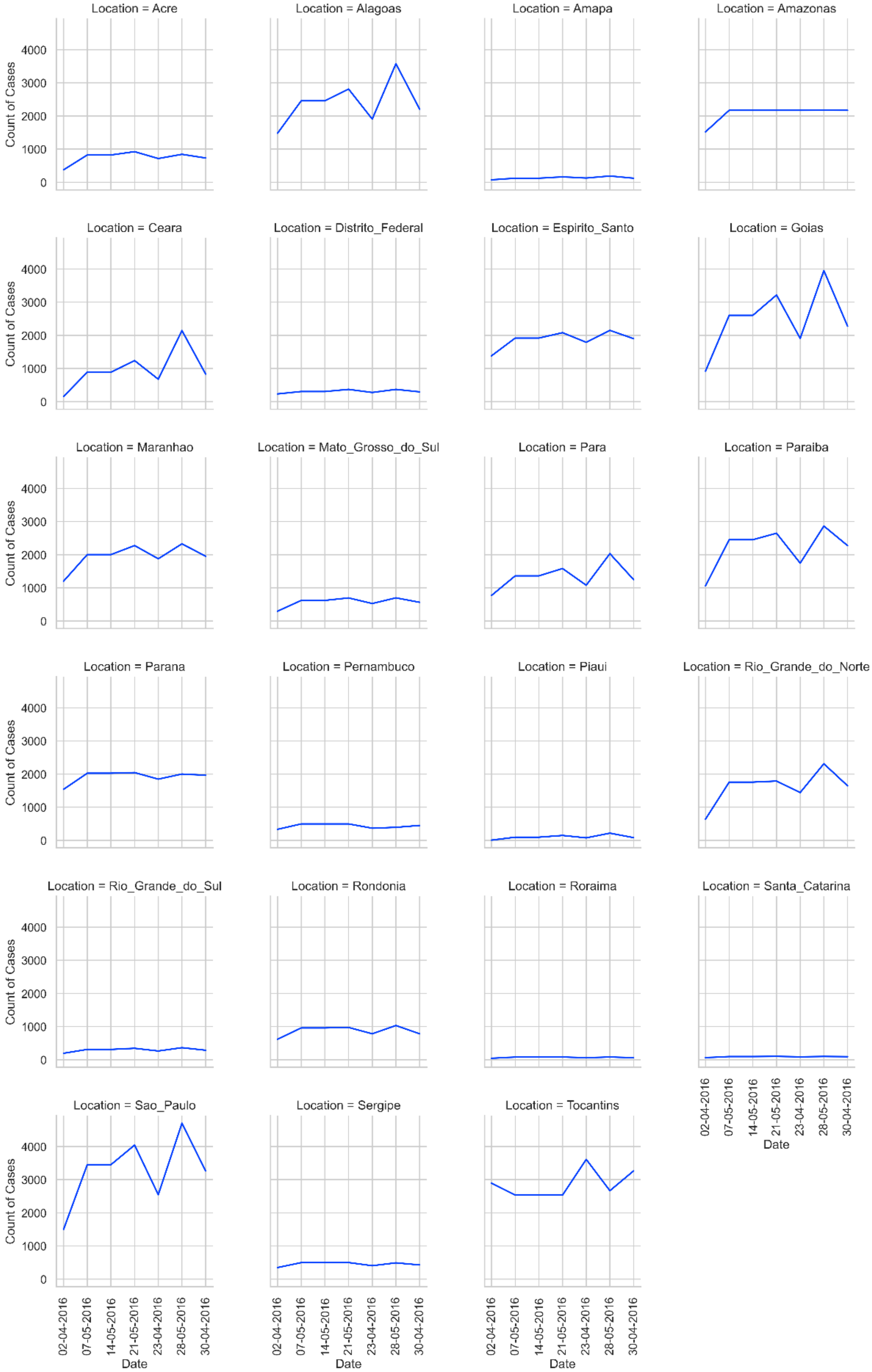


Figure 4.16 States with Lesser Spread of Infection

For the implementation of the proposed framework, data for each province is extracted for Brazil where each province act as one hexagon. Many data points which are required such as transmission mode and infection spread direction were lacking in the collected Zika virus dataset. These data points were synthetically generated using poison distribution. A portion of the modified dataset when imported in the Juypter notebook is shown in Figure 4.14. The trend of zika virus infection numbers in different provinces of Brazil are shown in Figure 4.15 and Figure 4.16. Table 4.8 shows all the inputs used for the training and testing of all the DNNs used in the proposed framework.

4.3.2.2 Training of DNNs

The synthetic dataset created, explained in previous section, is used to train, and test all the DNNs created in the proposed framework. Furthermore, the DNNs were trained in the sequence shown in Figure 4.9. The dataset was divided into two parts which are training dataset and validation dataset for evaluating the performance metrics. Basically, three DNNs were trained which are named type, direction, and marking for the brevity of experimental results. Type NN recommends the type of infection, direction NN provides the direction in which infection can spread and marking NN suggests the marking of a hexagon as red, orange, or green.

Table 4.9 Hyperparameters for the Training and testing of DNNs

No.	**Hyperparameters/Parameters**	**Type DNN**	**Marking DNN**	**Direction DNN**
1.	Metric	Accuracy	Accuracy	Accuracy
2.	Return type [State or Sequence]	Sequence	None	State
3.	Optimization Used	Short-Adam	Short-Adam	Adam
4.	Total Input Layer Neuron	5	3	2
5.	Hidden Layer Activation Function	ReLu	ReLu	ReLu
6.	Total Hidden Layers	1	4	6
7.	Output Layer Activation Function	Leaky ReLu	Sigmoid	Softmax
8.	Total Neuron in Output Layer	2	2	2
9.	Regularization	No	No	Dropout (0.2)
10.	Size of Mini-Batch	No	No	No
11.	Loss Function	Mean absolute error	Mean absolute error	Multiclass cross entropy
12.	Model	Sequential	Sequential	Sequential
13.	Type	Dense Long short-term memory	Dense BPNN	Dense Recurrent Neural Network

The hyperparameters which provided the best results are listed in Table 4.9 for all the DNNs. Figure 4.17, Figure 4.18, and Figure 4.19 provides the value of loss for type, direction and marking NNs respectively. Similarly, Figure 4.20, Figure 4.21, and Figure 4.22 provides the accuracy of type, direction and marking NN respectively for 300 epochs of training. An overall accuracy of 87% was achieved for the marking of the hexagons as compared with the ground truth labelling. Figure 4.23 shows the final marking predicted by the proposed framework for all the 22 hexagons.

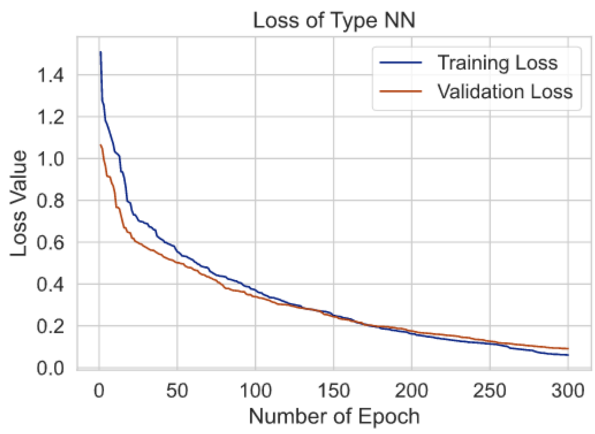


Figure 4.17 Type NN Loss

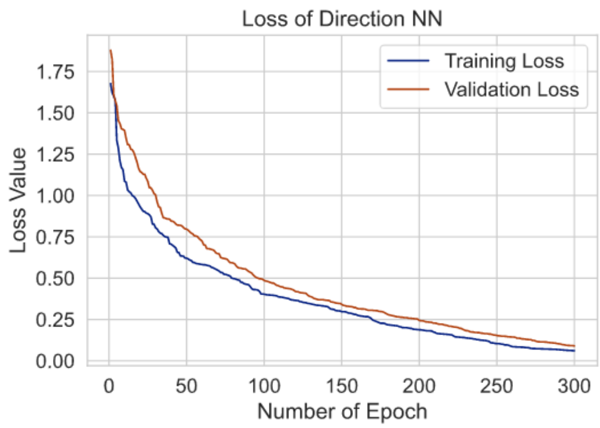


Figure 4.18 Direction NN Loss

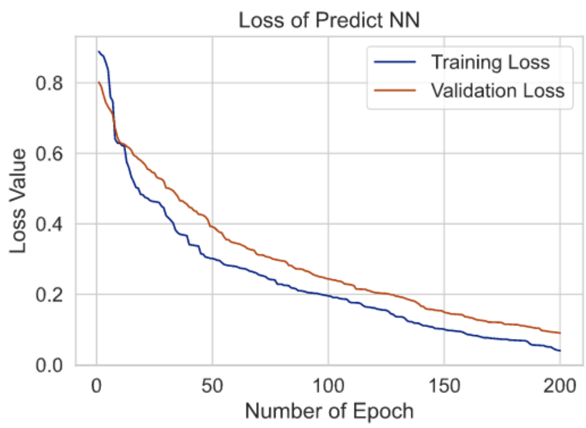


Figure 4.19 Marking NN Loss

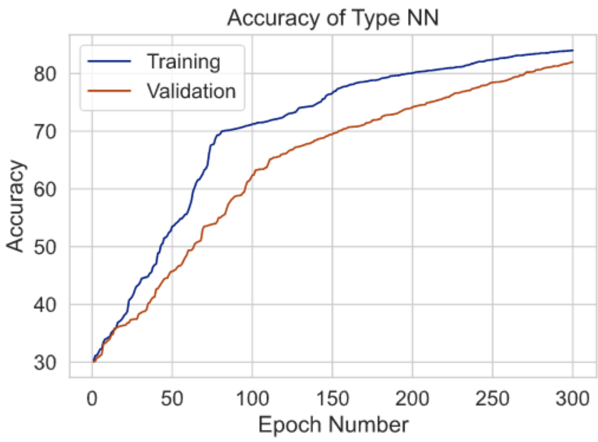


Figure 4.20 Type NN Accuracy

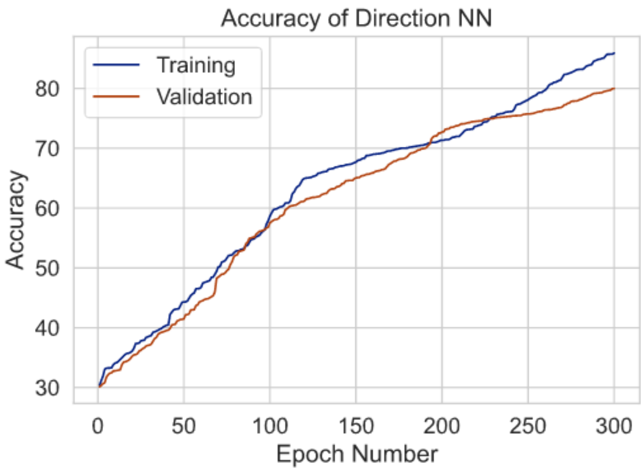


Figure 4.21 Direction NN Accuracy

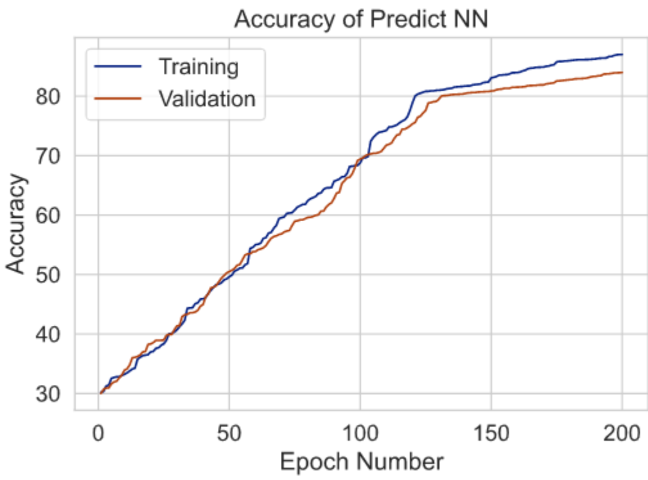


Figure 4.22 Marking NN Accuracy

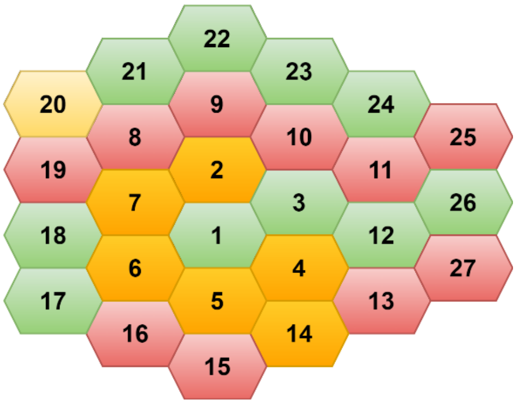


Figure 4.23 Predicted Marking of Hexagons

4.4 Conclusion

Many CARS uses combination of contextual and non-contextual data for generating user specific recommendations. However, the method adopted by traditional RS to first perform the context engineering and then to generate recommendations is not viable for today's CARS because of large amount and heterogenous data. DNNs can be used for these kinds of CARS because they can perform context engineering and generate recommendations as a single entity. Further, DNNs work better with large amount of data and can be used for latency sensitive applications also. This chapter have proposed three frameworks which uses contextual and non-contextual data parallelly to generate recommendations. Experimental evaluation of all the three proposed frameworks concluded that DNNs successfully generate highly accurate personalized recommendations. DNNs were also successful in generating time sensitive recommendations using transfer learning methodology as shown in second framework.

CHAPTER 5

CONTEXT RELATIONSHIPS AND SHARING USING DNNs

In the previous chapter, contextual and non-contextual input data was analysed using simple DNN and series of DNNs which generated highly accurate recommendations. However, frameworks discussed in the previous chapters were treated as a single entity and context relationship or context sharing was not performed by them. In any real-time CARS, context once generated by the sensor inputs may be needed to share with multiple CARS so that effective recommendations can be generated with minimum latency [5], [57], [120]. Furthermore, there can be relationship between multiple intermediate recommendations generated by DNNs which if modelled carefully can further enhance the recommendation accuracy [121], [122]. So, in this chapter a framework is designed which has multiple stages of recommendation generation and each stage share its output with other stages to generate the final recommendations. The output from these stages can be used by any number of CARS wherever these intermediate recommendations are relevant. The use case taken for the implementation of this framework is to colour mark any geographical area based on the spread of COVID-19. The recommendation is like one of the frameworks in the previous chapters, however, this framework has multiple stages to show that context generated from DNNs can be easily shared with other CARS.

The most effective method to stop the COVID-19 spread in its early stages was to impose lockdowns in multiple countries [123]. These lockdowns have serious implications on the economy of various nations, and it made difficult for many to even survive [124], [125]. Due to this the government started providing relaxations in the lockdown in a sequence. The unsystematic approach of relaxing the lockdown conditions is not a better method [126], [127]. A CARS framework must be developed which can take various contextual and non-contextual inputs and help the government for systematic relaxation of the lockdowns. These frameworks should be context sensitive because spread of any infectious virus is dynamic and a static machine learning system will not be able to adapt to change [120], [128]–[135]. In these types of frameworks, context can be population density, number of hospitals, number of doctors, infection density etc. These contextual parameters should be linked to a given geographical location. So, a common structure should be followed to divide any geographical regions. In the proposed framework, a hexagonal based structure is used, as it was done in one of the frameworks in the previous chapter.

5.1 The Proposed Framework

Figure 5.1 shows the proposed framework for effective sharing and relationship management of contextual and non-contextual data for COVID-19 recommendation generations. The DNN shown in Figure 5.1 is installed for all hexagons of any geographical region as shown in Figure 5.2. In the stage 1, the first NN predicts the amount of people arrive and leave a given hexagon. This would help in understanding the movement of people in a hexagon because higher the movement more would be the chances of infection spread. This NN is a CNN network which can take two types of input. In the first type, a camera can be installed in every entry and exit of a hexagon and second is the direct input on the number of people that enter or leave the hexagon. The camera feed will be processed by the CNN, and it will predict the number of people that leave or enter a given hexagon.

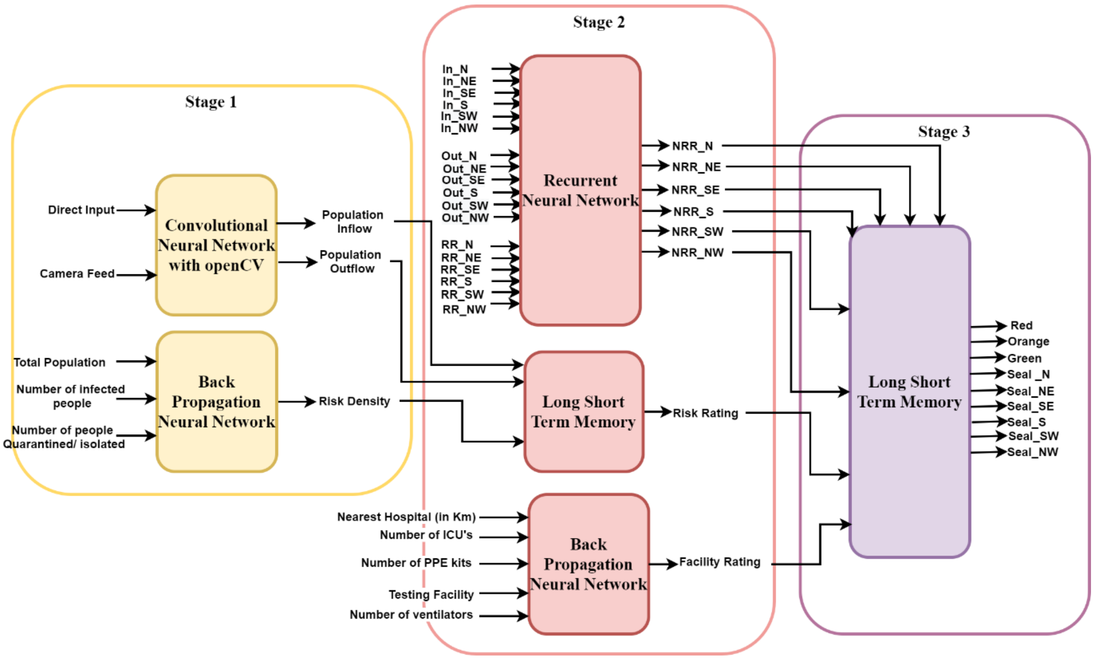


Figure 5.1 The Proposed Framework to Identify Contextual Relationships and for Contextual Sharing

The second NN in stage 1 is used to find the Risk Density (RD) of a hexagon which depends on three inputs as shown in Figure 5.1. RD means what is the possibility that a given hexagon will spread the virus in neighbouring hexagons based on its current state. In the stage 2, the top NN takes the value of total people moving in and out of the current hexagon neighbours as calculated in Stage 1, and Neighbor Risk Rating (NRR) of numeric type is predicted for all its neighbours. This would provide a value to current hexagon DNNs that how much it needs to protect itself from its neighbours. Every hexagon has six neighbours so this NN will produce six NRR values. The middle NN of Stage 2 will take long sequence of past values and predict the current risk rating of a hexagon. This NN takes a series of past values for current prediction, so it used LSTM for its effective working. The last NN in stage 2 predicts the Facility Rating (FR) for any hexagon using different inputs related to hospitals. FR is important because it predicts the medical strength of a given geographical area to effectively deal with the spread. For example, a hexagon with 1000 hospital beds can easily handle 1000 cases as compared to a hexagon with only 20 hospital beds. At the Stage 3, there is a single complex LSTM based NN which takes inputs from the Stage 2 and predicts the marking of a given hexagon. Also, it provides authorities with the recommendation to open or close boarders with neighbour hexagons. All the three stages are explained in detail in following sections.

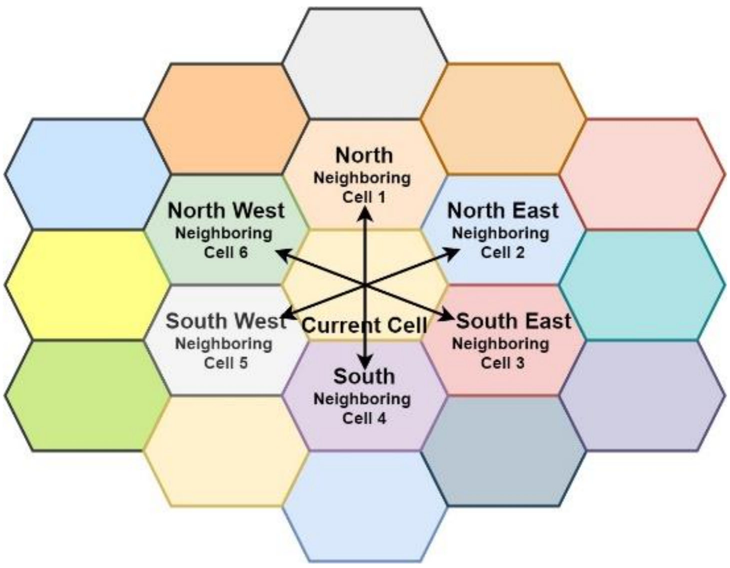


Figure 5.2 Hexagonal Structure for Geographical Area Classification

5.1.1 Inflow and outflow from a hexagon

A geographical area with heavy movement of persons is always a hotspot for larger spread of any infection. Thus, the governments must keep track of the movement of people from one hexagon to another. In the proposed framework, Algorithm 5.1 [136] is used which is CNN based NNs to count number of people crossing a given entry or exit point [137], [138]. Further, it may not be possible for an authority to install cameras in all the entry and exit points. In the proposed framework, the NN also takes a direct input which can be passed to the next stage. Table 5.1 shows the value of different notations used in the Figure 5.1 for this NN.

Algorithm 5.1: Inflow and outflow from a hexagon.

```
Mark a horizonal line at the entry/exit of the camera feed as a reference.
for Each element (people) in the frame assign a centroid do
    Mark a fix number to the new element.
    for every element do
        Find the Euclidean distance of every element centroid in the previous and the current frame.
        if Value is less than the threshold
            Indicate that element is moving and move the centroid.
        Elseif Value is greater than threshold
            Assign the element a new ID and treat it as new point
        else
            Delete the element from frame
    For every point, after any movement calculate the change in Y axis from the reference line
    If Value if negative
        Count this as inflow
    else
        Count it as Outflow
Continue with the feed.
```

Table 5.1 Inputs and Outputs to the CNN Block

Notation of Output	Explanation
In_N	Total people coming from North.
In_NW	Total people coming from North West.
In_SW	Total people coming from South West.
In_S	Total people coming from South.
In_SE	Total people coming from South East.
In_NE	Total people coming from North East.
Out_N	Total people going out towards north.
Out_NW	Total people going out towards North West.
Out_SW	Total people going out towards South West.
Out_S	Total people going out towards South.
Out_SE	Total people going out towards South East.
Out_NE	Total people going out towards North East.

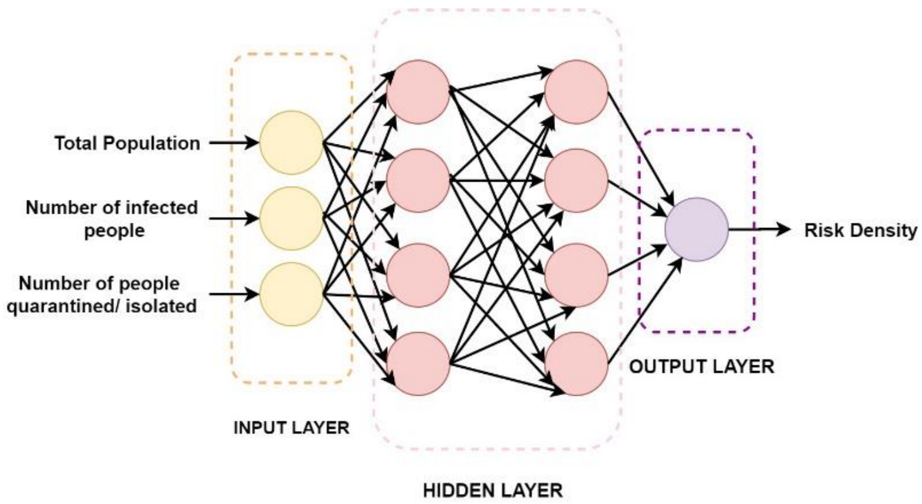


Figure 5.3 BPNN for Predicting RD

5.1.2 Risk Density Neural Network

Three inputs are used in this NN which recommends the risk density of a hexagon. The three inputs used are number of infectious cases, total population, and number of quarantined cases as depicted in Figure 5.3. The designed NN for the recommendation of risk density is shown in Figure 5.3 with input, hidden and output layers.

5.1.3 NRR using RNN

A sequence based RNN is used to recommend the NRR which can be used to detect the flow of infection as well as the risk a neighbour impose on a hexagon. All the input and output parameters used in this NN are shown in Figure 5.4. This RNN predicts the NRR for all of the hexagon neighbour. The main reason to use RNN in this NN is because past occurrence of the infectious disease should also be used in recommending the current NRR [139]. The designed NN takes input from the NN of Stage 1.

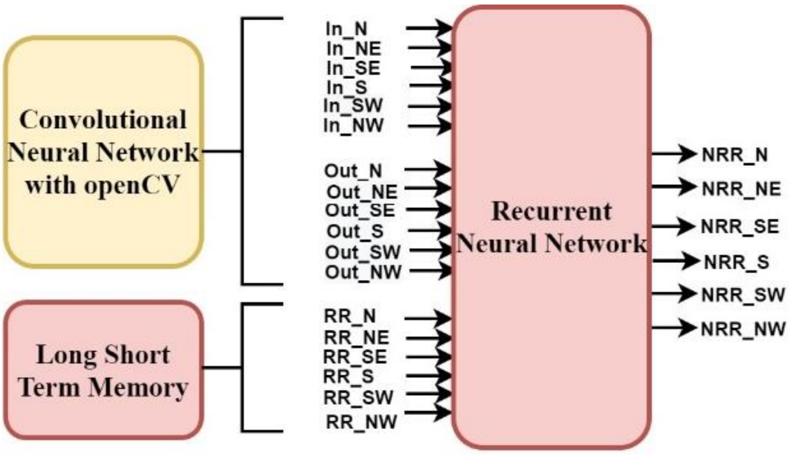


Figure 5.4 RNN to Calculate NRR from each Direction

5.1.4 LSTM based Risk Rating

Risk density has been already calculated in the Stage 1 of the proposed framework. However, risk rating also depends on the amount of inflow and outflow from a hexagon. For example, if there is a great amount of inflow from a high-risk density neighbour then it elevates the risk of spread of the infection. Furthermore, the calculation of current risk rating should also consider a long sequence of previous risk rating values for better recommendations. So, the LSTM is better suitable in this DNN for better predictions. The RR NN inputs are shown in Figure 5.5.

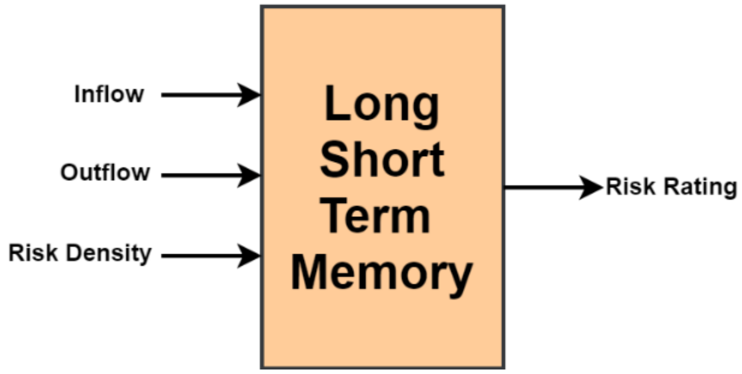


Figure 5.5 RR Input and Output

5.1.5 Facility Rating

The decision to impose or relax the lockdown in a hexagon also depends on the quantity and quality of the medical services in that hexagon. In the proposed framework, a simple BPNN is used which takes inputs related to medical facilities and provide a facility rating between 1 to 10 for every hexagon. The designed NN structure is shown in Figure 5.6.

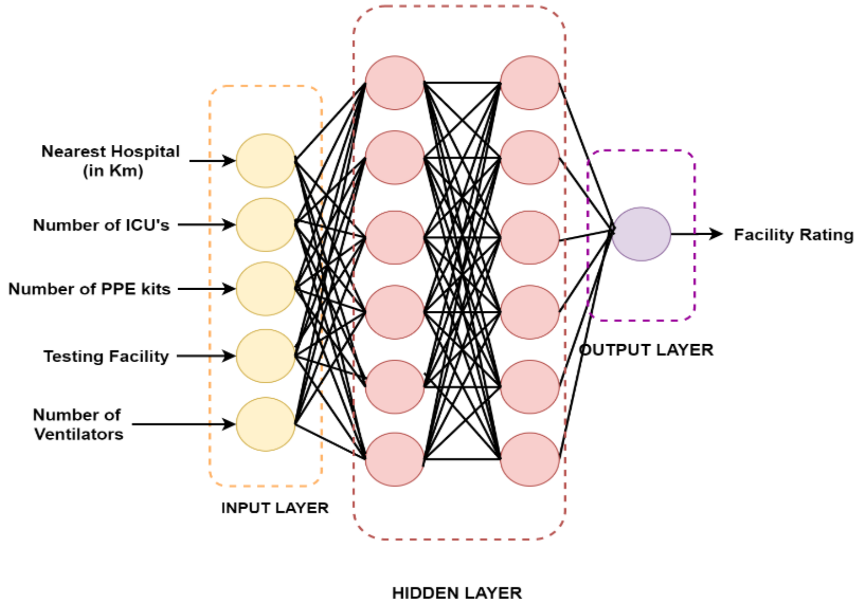


Figure 5.6 BPNN for Generating FR of a Hexagon

5.1.6 Colour Marking using LSTM

All the recommendations generated by CARS of Stage 2 are used by a single LSTM network for generating the final recommendations. Figure 5.7 illustrates the NN architecture used in this stage. It is a complex and deep network with four hidden layers containing eight LSTM neurons each. The output of this NN is the colour marking of the hexagon with respective colours as red, orange, and green and a binary value stating whether inflow or outflow is allowed from a neighbour. Table 5.2 lists different notations used for this NN as were shown in Figure 5.1.

Table 5.2 Notations used in Figure 5.1

Explanation	Notation of Output
Declare the hexagon as emergency area	Red
Declare the hexagon as sensitive area	Orange
Declare the hexagon as risk free area	Green
Seal the North Boarder	Seal_N
Seal the Northeast Boarder	Seal_NE
Seal the Southeast Boarder	Seal_SE
Seal the South Boarder	Seal_S
Seal the Southwest Boarder	Seal_SW
Seal the Northwest Boarder	Seal_NW

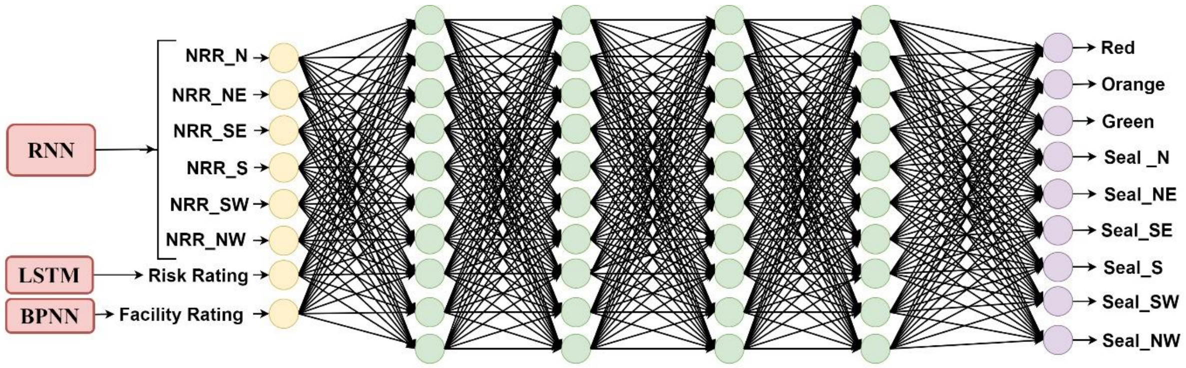


Figure 5.7 LSTM to Mark the Sensitivity of an Area

5.2 Performance Analysis of Proposed Framework

To understand the contextual sharing and relationships in the proposed framework for COVID-19 area marking, internet was searched thoroughly to find a dataset with all the contextual and non-contextual data points. The required dataset was not available on the internet, so a hybrid synthetic dataset is generated by taking values from various sources. In the synthetic generated dataset, a structure with 22 hexagons were considered representing 22 districts of Haryana state in India with ten months of data points. As the proposed framework contains multiple DNNs, these DNNs were individual trained and tested for their respective datasets which is discussed in detail in this section. The required data was collected from [140], [141] some of the data was also generated using various distributions like poison, normal and gaussian [142].

5.2.1 Synthetic Dataset Generation

As discussed in the previous section, a camera feed can be used to count inflow and outflow of people from a given hexagon. Counting of persons using camera feed is not the main objective of this paper so for the brevity and proof of concept direct input of inflow and outflow was considered. A poison distribution based random values were generated for both inflow and outflow of 22 hexagons with their respective neighbour. The constant rate of 1500 persons arrival and leaving any hexagon is considered for the synthetic data generation. Figure 5.11 shows the respective heatmap created for this data point. For the population density data, Haryana government website census data was used. For other data points like active cases, recovered cases, number of hospitals, number of beds in hospitals etc. covidindia.org website was used [143], snapshot of the collected data is shown in Figure 5.8. The Figure 5.9 depicts various data points of the Kurukshetra district of Haryana for the ten months. Similarly, these data points were also collected for rest of the 21 districts in consideration. Figure 5.10 shows the number of identified cases for 20 hexagons considered.

	District Name	hexagon_no.	Total Inflow	Total Outflow	IN_N	IN_NE	IN_SE	IN_S	IN_SW	IN_NW	...	ICU_BED	TESTING_LABS	NO_VENTILATOR	LABEL
0	Ambala	1	8441	11356	772	1737	462	1699	1428	2343	...	120	2.0	16	Green
1	Bhiwani	2	11066	8878	1679	1011	2938	322	2934	2182	...	180	NaN	24	Green
2	Charkhi Dadri	3	8931	7613	2254	239	2022	2645	1036	735	...	30	NaN	4	Green
3	Faridabad	4	7645	10100	1393	2718	780	683	212	1859	...	90	4.0	12	Red
4	Fatehabad	5	7999	7050	429	2716	204	2188	1133	1329	...	30	NaN	4	Green
5	Gurugram	6	10794	6759	2672	313	2814	1186	2261	1548	...	60	13.0	8	Red
6	Hissar	7	10859	5809	1090	1560	411	2168	2842	2788	...	75	3.0	10	red
7	Jhajjar	8	7555	11413	274	1521	586	2335	642	2197	...	165	1.0	22	Green
8	Jind	9	9913	6969	2463	2396	1926	1206	1021	901	...	75	1.0	10	Green
9	Kaithal	10	7004	11834	1235	2297	349	962	604	1557	...	30	NaN	4	Green

Figure 5.8 The Hybrid Dataset

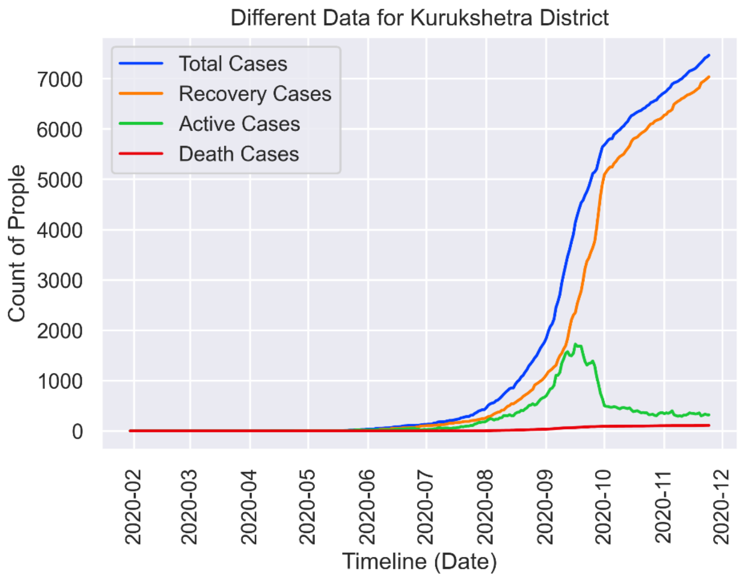


Figure 5.9 Trend of Cases in Kurukshetra District

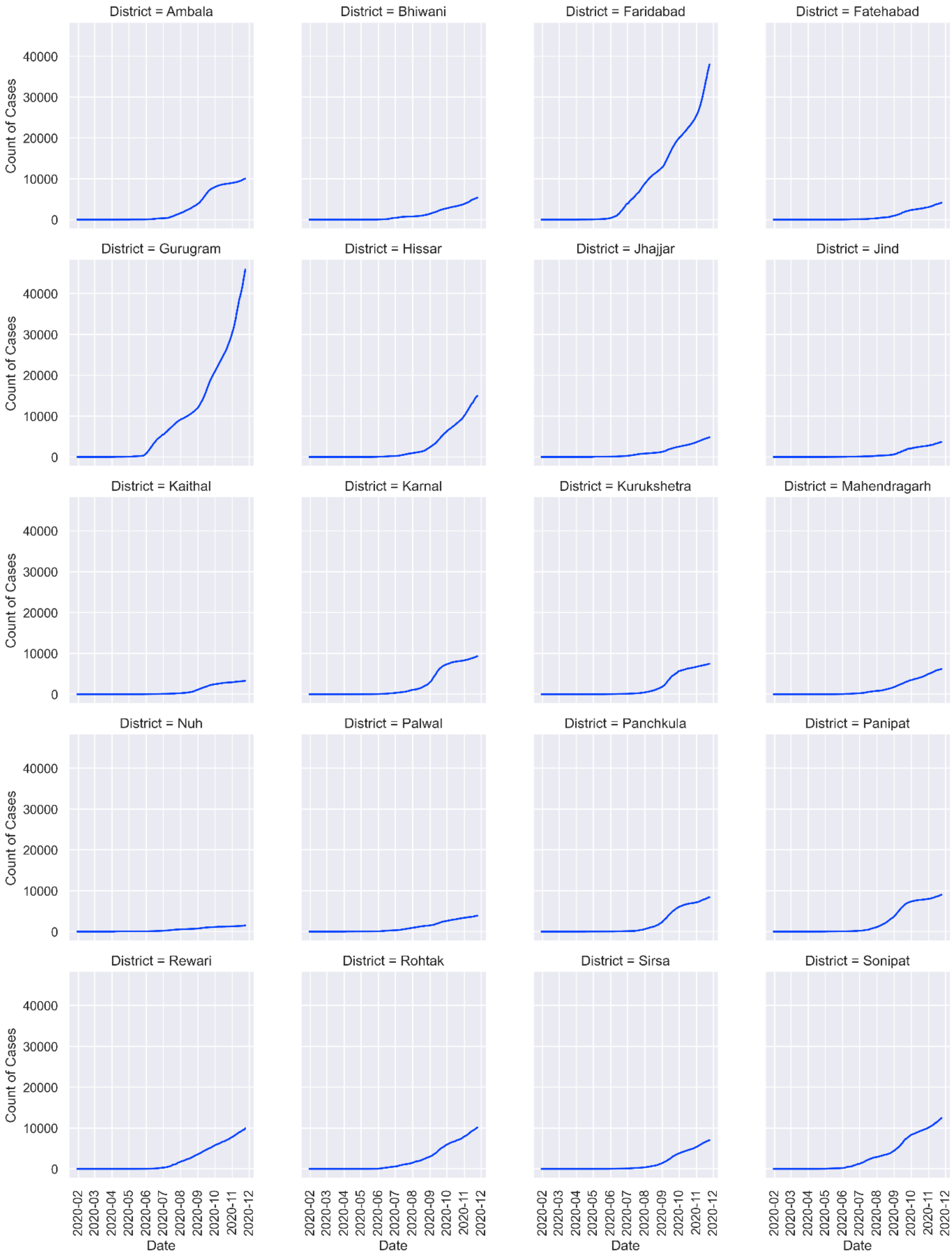


Figure 5.10 Number of Identified Cases in 20 Districts of Haryana

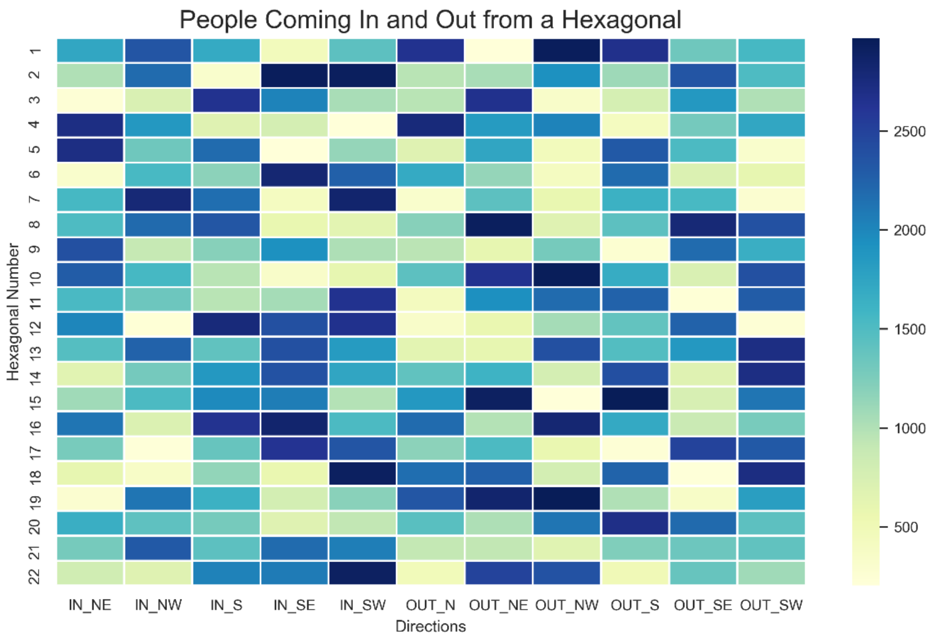


Figure 5.11 Heatmap of Movement of People from One Hexagon to Another

Table 5.3 Hyperparameters used in Various DNNs

S. No.	Parameter	Risk Density	Neighbor risk rating	Risk rating	Facility rating	Final
1.	Type	Dense BPNN	Dense Recurrent neural network	Dense Long short-term memory	Dense BPNN	Dense Long short-term memory
2.	Model	Sequential	Sequential	Sequential	Sequential	Sequential
3.	Total Input Layer Neuron	3	18	3	5	8
4.	Total Neuron in Output Layer	1	6	1	1	9
5.	Total Hidden Layers	2 (4 Neuron Each)	4	1	2 (6 Neuron Each)	4 (10 neuron Each)
6.	Hidden Layer Activation Function	ReLu	ReLu	ReLu	ReLu	ReLu
7.	Output Layer Activation Function	Leaky ReLu	Softmax	Leaky ReLu	Sigmoid	Softmax
8.	Optimization Used	Short-Adam	Adam	Short-Adam	Short-Adam	Adam
9.	Regularization	No	Dropout (0.2)	No	No	Dropout (0.2)
10.	Mini-Batch Size	No	No	No	No	No
11.	Loss Function	Mean absolute error	Multiclass cross entropy	Mean absolute error	Mean absolute error	Multiclass cross entropy
12.	Metric	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy
13.	Return type	No	State	Sequence	No	State

5.2.2 DNNs Training

The synthetic dataset created in the previous section for 22 hexagons was used for all the DNNs in the desired sequence depicted in Figure 5.1. The hyperparameters for which each DNN provided the best result are listed in Table 5.3. Figure 5.12, Figure 5.13, Figure 5.14, Figure 5.15, and Figure 5.16 shows the increase in accuracy and decrease in loss for all the DNNs when trained with the synthetic dataset. After all the individual training and testing, final colour marking for all the 22 hexagons were predicted which is shown in Figure 5.17.

Python 3.7 and Tensorflow 2.0 [144], [145] is used with other required packages like Keras, NumPy, pandas, seaborn, matplotlib etc. for the implementation of the DNNs. The performance metrics evaluated from the experimental setup are listed in Table 5.4. The accuracy was taken as base metric for the evaluation of different DNNs because the dataset created does not have any class imbalance problem. All the performance metrics provided good results for all the DNN's where the final marking got the accuracy of 83% when compared with the ground truth labelling. The other parameters like precision, recall and F-measure [146] also provided score above 0.8 which is good.

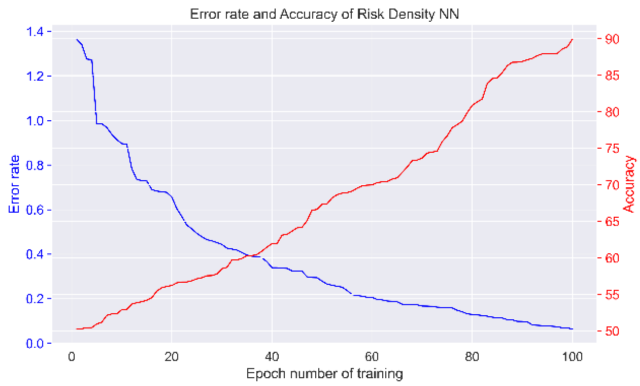


Figure 5.12 RD Error rate and Accuracy

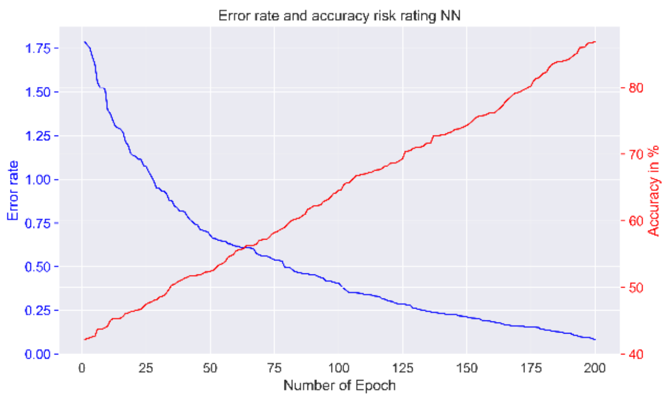


Figure 5.13 RR Error rate and Accuracy

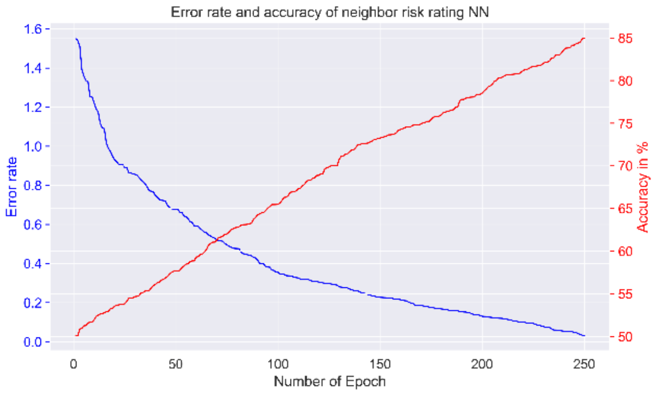


Figure 5.14 NRR Error rate and Accuracy

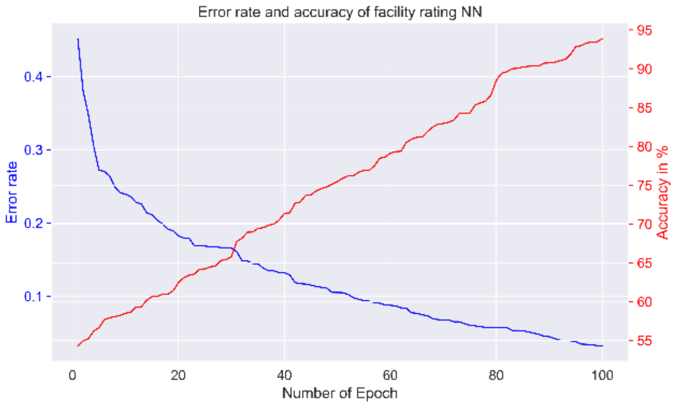


Figure 5.15 FR Error rate and Accuracy

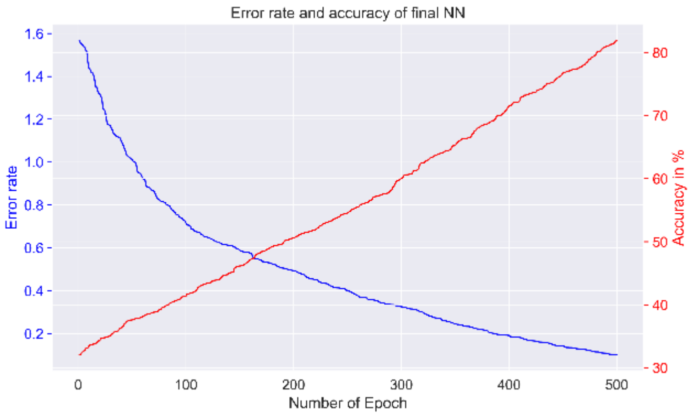


Figure 5.16 Final NN Error rate and Accuracy

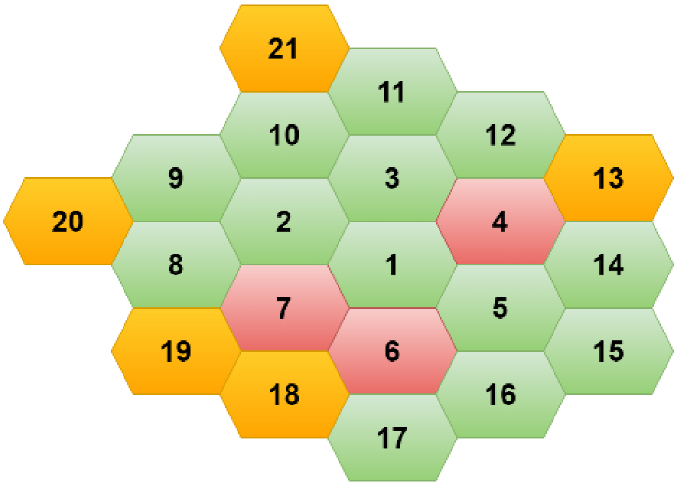


Figure 5.17 Colour Marking of Hexagons

Table 5.4 Performance Metrics of All DNNs

S. No.	Parameter	RD	NRR	RR	FR	Final
1.	Accuracy	0.91	0.85	0.87	0.93	0.83
2.	Precision	0.92	0.86	0.88	0.93	0.84
3.	Recall	0.94	0.90	0.92	0.96	0.88
4.	F-Measure	0.93	0.88	0.90	0.95	0.86

5.2.3 Performance comparison with variations

After searching the internet thoroughly, no similar framework was found to compare the performance of the proposed framework. Basic objective of the proposed framework is that the context can be shared with different CARS framework without any degradation of the performance. Further, it was hypothesized that different frameworks of DNN like CNN, RNN, LSTM if used based on their respective type of data input, they can formulate better context relationship than a single type of DNN like BPNN. So, the proposed framework is compared with two variations keeping the above said objective in mind.

- **Single NN:** A single DNN is used in which all inputs of the proposed framework are used, and it predicts the marking of the hexagon. In this variation, the single DNN used is LSTM and its training parameters are provided in Table 5.5.
- **BPNN:** multi-stage framework is kept in this variation, however, all the internal DNNs are replaced with BPNN. Table 5.5 has all the hyperparameters used in the training of this variation of network.

Figure 5.18 depicts the comparison of the above said variations of the proposed framework. As hypothesised the variation performed way less than the proposed framework in all the performance metrics used.

Table 5.5 Risk Density NN Parameters

S. No.	Parameter	Variation 1	Variation 2				
			RD	**NRR**	**RR**	**FR**	**Final**
1.	Type	Dense LSTM	Dense BP NN	Dense BP NN	Dense BP NN	Dense BP NN	Dense BP NN
2.	Model	Sequential	Sequential	Sequential	Sequential	Sequential	Sequential
3.	Number of Input Neuron	28	3	18	3	5	8
4.	Number of Output Neuron	9	1	6	1	1	9
5.	Number of Hidden Layers	6 (15 neuron Each)	2 (4 Neuron Each)	2 (4 Neuron Each)	2 (4 Neuron Each)	2 (6 Neuron Each)	2 (4 Neuron Each)
6.	Activation Function in Hidden Layer	ReLu	ReLu	ReLu	ReLu	ReLu	ReLu
7.	Output Layer Activation Function	Softmax	Leaky ReLu	Leaky ReLu	Leaky ReLu	Sigmoid	Softmax
8.	Optimization	Adam	Short Adam	Short Adam	Short Adam	Short Adam	Short Adam
9.	Regularization Technique	Dropout (0.2)	None	None	None	None	None
10.	Mini-Batch Size	None	None	None	None	None	None
11.	Loss Function	MCE	MAE	MAE	MAE	MAE	MAE
12.	Metric	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy
13.	Return type [State or Sequence]	State	None	None	None	None	None

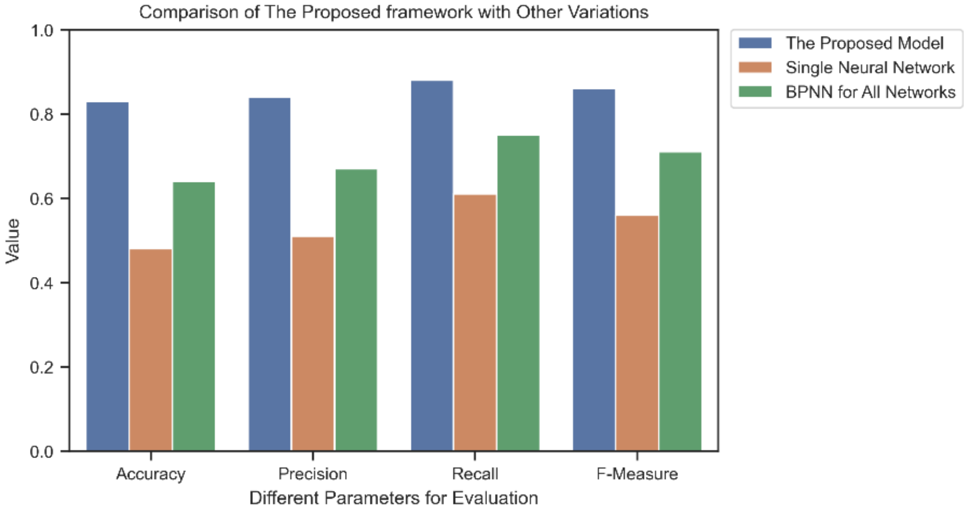


Figure 5.18 Comparison of the Proposed Framework with its Variation

5.2.4 Discussion of experimental results

Contextual parameters used in the proposed framework are:

- Geographical location in the form of hexagons.
- Density of infection

- Population
- Inflow and outflow
- Medical facilities

Every hexagon representing a specific geographical area has all the above said contextual data unique to it. In the proposed framework, a user can add more contextual data based on the requirement of the applications and recommendations. The proposed framework's experimental setup studied the relevancy of context relationship inside a CARS (represented as stage in this framework) and context sharing based on the required analysed context values. The proposed framework was simulated for an average size dataset of 22 hexagons with data collected from various sources. The proposed framework can be easily scaled up to a nation level because a hexagon DNN will be of same size and complexity. Lastly, the comparison of the proposed framework with its two variations also stated the hypothesis that context sharing, and relationship results in better recommendation generations.

5.3 Conclusion

In this chapter, the objective was to design a DNN based framework which can manage the contextual relationship inside a CARS and can share the intermediate contextual recommendations with other CARS to generate time sensitive recommendations. Different CARS are represented as stages in the proposed framework and a geographical location marking framework was developed based on the COVID-19 dataset. Implementation of the proposed framework with hybrid dataset and its comparison with different variations proved the hypothesis. The CARS with multiple stages performed better than its alternatives in all the performance metrics used.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

CARS are the future to generate personalized recommendations, however, to analyze the large amount of data and recommending user specific predictions is a difficult task. Furthermore, CARS works in two steps, first it performs context engineering and, the second is to generate recommendations. DNNs have been proposed to analyze large amount of contextual data and to perform context engineering and generate predictions as a single unit. Also, DNNs were able to effectively learn patterns and generate effective recommendations based on all types of context input. Moreover, DNNs can effectively transform raw contextual data to low dimensional space while retaining the meaningful information. Further, DNN based frameworks can be implemented in any application domains, but specialized framework is required to design for different application areas. However, transfer learning can be used to effectively design more frameworks in the similar application areas. To work a CARS better, analyzing the contextual relationship and its further sharing is a crucial task which was effectively done using DNNs.

6.2 Summary of Major Contribution

In this thesis, research objectives were formed with the intention to improve the recommendation and its process in CARS. Systemic design of framework was conducted to achieve all the objectives stated in Chapter 1. Some of the notable contributions of this thesis are:

i. With large number of sensors, contextual inputs will have large number of attributes which may be contradictory or complementary to each other. So, a framework is proposed which uses DNNs to reduce the dimension of input contextual data by learning complex patterns in the input data.

ii. Multiple frameworks are proposed which takes both contextual and non-contextual data and generate user specific recommendation without the need of separate context engineering.

iii. Context relationship with other data values and its sharing is studied in a framework which has multiple stages using DNNs.

6.3 Future Scope

In this thesis, the context considered is collected in the form of structured data. However, in current era a better context can be collected from unstructured data like audio, video collected from heterogenous sources and may be in different formats. Analysis and context engineering of these contextual data may be the better path for the future scope of CARS. Further, more generalized frameworks can be designed which can be effectively used for transfer learning in multiple application areas. Moreover, neural turing machines can be used to improve the stability of NNs. Also, context can be analyzed based upon its classification, like

- Natural Context
- Human Context
- Action Context

Lastly, the deployment of these frameworks on different types of computing infrastructure also has great scope in the future.

REFERENCES

[1] M. M. Theimer and B. N. Schilit, “Disseminating Active Map Information to Mobile Hosts,” *IEEE Netw.*, vol. 8, no. 5, pp. 22–32, 1994, doi: 10.1109/65.313011.

[2] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, “Smart community: An internet of things application,” *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 68–75, Nov. 2011, doi: 10.1109/MCOM.2011.6069711.

[3] A. K. Dey and G. Abowd, “Towards a Better Understanding of Context and Context-Awareness,” in *Handheld and Ubiquitous Computing*, Springer Berlin Heidelberg, 1999, pp. 304–307.

[4] “A brief description of Contextual Factors | by Sebastian Amenabar | Medium.” https://medium.com/@sebamenabar/a-brief-description-of-contextual-factors-441ed2601ca7 (accessed Apr. 26, 2022).

[5] M. Unger, A. Bar, B. Shapira, and L. Rokach, “Towards latent context-aware recommendation systems,” *Knowledge-Based Syst.*, vol. 104, pp. 165–178, Jul. 2016, doi: 10.1016/j.knosys.2016.04.020.

[6] F. Xhafa and A. Abraham, “Meta-heuristics for grid scheduling problems,” *Stud. Comput. Intell.*, vol. 146, pp. 1–37, 2008, doi: 10.1007/978-3-540-69277-5_1.

[7] M. Kalra and S. Singh, “A review of metaheuristic scheduling techniques in cloud computing,” *Egypt. Informatics J.*, vol. 16, no. 3, pp. 275–295, Nov. 2015, doi: 10.1016/j.eij.2015.07.001.

[8] V. Sharma and A. K. Tripathi, “A systematic review of meta-heuristic algorithms in IoT based application,” *Array*, vol. 14, p. 100164, Jul. 2022, doi: 10.1016/J.ARRAY.2022.100164.

[9] M. Bansal, A. Goyal, and A. Choudhary, “A comparative analysis of K-Nearest Neighbour, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning,” *Decis. Anal. J.*, p. 100071, May 2022, doi: 10.1016/J.DAJOUR.2022.100071.

[10] G. Cohen, G. Sapiro, and R. Giryes, “DNN or k-NN: That is the Generalize vs. Memorize Question,” May 2018, doi: 10.48550/arxiv.1805.06822.

[11] C. C. Aggarwal, *Neural Networks and Deep Learning - A Textbook*. 2018.

[12] J. Patterson and A. Gibson, *Deep Learning A Practitioner’s Approach*. 2005.

[13] R. Vidal, J. Bruna, R. Giryes, and S. Soatto, “Mathematics of Deep Learning,” Dec. 2017, doi: 10.48550/arxiv.1712.04741.

[14] J. Patterson and A. Gibson, *Deep Learning A practitioner's approach*. 2008.

[15] R. Mittal, S. Arora, V. Bansal, and M. P. S. Bhatia, "An Extensive Study on Deep Learning: Techniques, Applications," *Arch. Comput. Methods Eng.*, vol. 28, no. 7, pp. 4471–4485, Dec. 2021, doi: 10.1007/s11831-021-09542-5.

[16] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414–454, Mar. 2014, doi: 10.1109/SURV.2013.042313.00197.

[17] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin, "Context-Aware Recommender Systems," *AI Mag.*, vol. 32, no. 3, pp. 67–80, 2011, doi: 10.1609/AIMAG.V32I3.2364.

[18] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Wmcsa*, 1899, pp. 85–90, doi: 10.1109/MCSA.1994.512740.

[19] K. Henricksen, *A framework for context-aware pervasive computing applications*. 2003.

[20] A. H. Van Bunningen, L. Feng, and F. M. G. Apers, "Context for ubiquitous data management," in *Proceedings of the International Workshop on Ubiquitous Data Management , UDM 2005*, 2005, pp. 17–24, doi: 10.1109/UDM.2005.7.

[21] M. Chantzara and M. Anagnostou, "Evaluation and Selection of Context Information," 2005.

[22] A. Ferscha, S. Vogl, and W. Beer, "Context Sensing, Aggregation, Representation and Exploitation in Wireless Networks," *Scalable Comput. Pract. Exp.*, vol. 6, no. 2, pp. 71–81, 2005.

[23] K. Wrona and L. Gomez, "Context-aware security and secure context-awareness in ubiquitous computing environments," in *Proceedings of the XXI Autumn Meeting of Polish Information Processing Society*, 2005, pp. 255–265, doi: 10.1.1.96.6761.

[24] G. Hynes, V. Reynolds, and M. Hauswirth, "A context lifecycle for web-based context management services," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5741 LNCS, Springer Berlin Heidelberg, 2009, pp. 51–65.

[25] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, p. 263, 2007, doi: 10.1504/IJAHUC.2007.014070.

[26] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakraborty, "Intelligent agents meet the semantic Web in smart spaces," *IEEE Internet Comput.*, vol. 8, no. 6, pp. 69–

79, 2004, doi: 10.1109/MIC.2004.66.

[27] J. Indulska and P. Sutton, “Location management in pervasive systems,” in *Proceedings of the Australasian information security workshop conference on ACSW frontiers*, 2003, vol. 34, no. vi, pp. 143–151.

[28] A. Schmidt and K. Van Laerhoven, “How to build smart appliances?,” *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 66–71, 2001, doi: 10.1109/98.944006.

[29] M. Knappmeyer, S. L. Kiani, C. Frà, B. Moltchanov, and N. Baker, “ContextML: A light-weight context representation and context management schema,” *ISWPC 2010 - IEEE 5th Int. Symp. Wirel. Pervasive Comput. 2010*, pp. 367–372, 2010, doi: 10.1109/ISWPC.2010.5483753.

[30] S. Yanwei, Z. Guangzhou, and P. Haitao, “Research on the context model of intelligent interaction system in the Internet of Things,” in *IT in Medicine and Education (ITME), 2011 International Symposium on*, Dec. 2011, vol. 2, pp. 379–382, doi: 10.1109/ITiME.2011.6132129.

[31] T. Van Nguyen, W. Lim, H. A. Nguyen, and D. Choi, “Context awareness framework based on contextual graph,” 2008, doi: 10.1109/WOCN.2008.4542542.

[32] H. Myeong, J. Y. Chang, and K. M. Lee, “Learning object relationships via graph-based context model,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2727–2734, doi: 10.1109/CVPR.2012.6247995.

[33] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist: Effective modeling in RDFS and OWL*. 2011.

[34] A. Sheth, C. Henson, and S. S. Sahoo, “Semantic sensor web,” *IEEE Internet Comput.*, vol. 12, no. 4, pp. 78–83, 2008, doi: 10.1109/MIC.2008.87.

[35] L. Yu, *A Developer’s Guide to the Semantic Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[36] B. Y. Lim and A. K. Dey, “Toolkit to support intelligibility in context-aware applications,” *UbiComp’10 - Proc. 2010 ACM Conf. Ubiquitous Comput.*, pp. 13–22, 2010, doi: 10.1145/1864349.1864353.

[37] W. Y. B. Lim *et al.*, “When Information Freshness Meets Service Latency in Federated Learning: A Task-Aware Incentive Scheme for Smart Industries,” *IEEE Trans. Ind. Informatics*, vol. 18, no. 1, pp. 457–466, Jan. 2022, doi: 10.1109/TII.2020.3046028.

[38] D. Gil, A. Ferr??ndez, H. Mora-Mora, and J. Peral, “Internet of things: A review of

surveys based on context aware intelligent services," *Sensors (Switzerland)*, vol. 16, no. 7, p. 1069, Jul. 2016, doi: 10.3390/s16071069.

[39] A. Q. Gill, N. Phennel, D. Lane, and V. L. Phung, "IoT-enabled emergency information supply chain architecture for elderly people: The Australian context," *Inf. Syst.*, vol. 58, pp. 75–86, Jun. 2016, doi: 10.1016/j.is.2016.02.004.

[40] A. Abbas, L. Zhang, and S. U. Khan, "A survey on context-aware recommender systems based on computational intelligence techniques," *Computing*, vol. 97, no. 7, pp. 667–690, Jul. 2015, doi: 10.1007/s00607-015-0448-7.

[41] Y. Chen, J. Zhou, and M. Guo, "A context-aware search system for Internet of Things based on hierarchical context model," *Telecommun. Syst.*, vol. 62, no. 1, pp. 77–91, May 2016, doi: 10.1007/s11235-015-9984-x.

[42] N. Sachdeva, R. Dhir, and A. Kumar, "Empirical analysis of Machine Learning Techniques for context aware Recommender Systems in the environment of IoT," in *Proceedings of the International Conference on Advances in Information Communication Technology & Computing - AICTC '16*, 2016, pp. 1–7, doi: 10.1145/2979779.2979818.

[43] P. Du, P. Putra, S. Yamamoto, and A. Nakao, "A context-aware IoT architecture through software-defined data plane," in *Proceedings - 2016 IEEE Region 10 Symposium, TENSYMP 2016*, May 2016, pp. 315–320, doi: 10.1109/TENCONSpring.2016.7519425.

[44] S. A. A. Amin, A. Ali-Eldin, and H. A. Ali, "A context-aware dispatcher for the Internet of Things: The case of electric power distribution systems," *Comput. Electr. Eng.*, vol. 52, pp. 183–198, 2016, doi: 10.1016/j.compeleceng.2015.05.012.

[45] M. Khan, S. Din, S. Jabbar, M. Gohar, H. Ghayvat, and S. C. Mukhopadhyay, "Context-aware low power intelligent SmartHome based on the Internet of things," *Comput. Electr. Eng.*, vol. 52, pp. 208–222, May 2016, doi: 10.1016/j.compeleceng.2016.04.014.

[46] R. Sandhu and S. K. Sood, "A stochastic game net-based model for effective decision-making in smart environments," *Concurr. Comput. Pract. Exp.*, 2016, doi: 10.1002/cpe.3843.

[47] C. Kamienski *et al.*, "Application development for the Internet of Things: A context-aware mixed criticality systems development platform," *Comput. Commun.*, vol. 104, pp. 1–16, May 2017, doi: 10.1016/j.comcom.2016.09.014.

[48] S. A. Rokni and H. Ghasemzadeh, "Autonomous sensor-context learning in dynamic human-centered internet-of-things environments," in *Proceedings of the 35th*

International Conference on Computer-Aided Design - ICCAD '16, 2016, pp. 1–6, doi: 10.1145/2966986.2967008.

[49] I. Ben Sassi, S. Mellouli, and S. Ben Yahia, "Context-aware recommender systems in mobile environment: On the road of future research," *Inf. Syst.*, vol. 72, pp. 27–61, Dec. 2017, doi: 10.1016/j.is.2017.09.001.

[50] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, "A hybrid recommender system using artificial neural networks," *Expert Syst. Appl.*, vol. 83, pp. 300–313, Oct. 2017, doi: 10.1016/j.eswa.2017.04.046.

[51] D. Kim, C. Park, J. Oh, and H. Yu, "Deep hybrid recommender systems via exploiting document context and statistics of items," *Inf. Sci. (Ny).*, vol. 417, pp. 72–87, Nov. 2017, doi: 10.1016/j.ins.2017.06.026.

[52] Y. Liu, B. Liu, L. Shan, and X. Wang, "Modelling context with neural networks for recommending idioms in essay writing," *Neurocomputing*, vol. 275, pp. 2287–2293, Jan. 2018, doi: 10.1016/j.neucom.2017.11.005.

[53] N. M. Villegas, C. Sánchez, J. Díaz-Cely, and G. Tamura, "Characterizing context-aware recommender systems: A systematic literature review," *Knowledge-Based Syst.*, vol. 140, pp. 173–200, Jan. 2018, doi: 10.1016/J.KNOSYS.2017.11.003.

[54] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network," *BMC Med. Res. Methodol.*, vol. 18, no. 1, pp. 1–12, Feb. 2018, doi: 10.1186/s12874-018-0482-1.

[55] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 1–37, Jun. 2019, doi: 10.1007/s10462-018-9654-y.

[56] Z. Liu, Q. Liu, W. Xu, Z. Liu, Z. Zhou, and J. Chen, "Deep learning-based human motion prediction considering context awareness for human-robot collaboration in manufacturing," *Procedia CIRP*, vol. 83, pp. 272–278, Jan. 2019, doi: 10.1016/j.procir.2019.04.080.

[57] S. Raza and C. Ding, "Progress in context-aware recommender systems - An overview," *Comput. Sci. Rev.*, vol. 31, pp. 84–97, Feb. 2019, doi: 10.1016/j.cosrev.2019.01.001.

[58] P. Pradeep and S. Krishnamoorthy, "The MOM of context-aware systems: A survey," *Comput. Commun.*, vol. 137, pp. 44–69, Mar. 2019, doi: 10.1016/j.comcom.2019.02.002.

[59] C. Djellali and M. Adda, “A new hybrid deep learning model based-recommender system using artificial neural network and hidden Markov model,” *Procedia Comput. Sci.*, vol. 175, pp. 214–220, Jan. 2020, doi: 10.1016/j.procs.2020.07.032.

[60] J. Wang, L. Zhu, T. Dai, and Y. Wang, “Deep memory network with Bi-LSTM for personalized context-aware citation recommendation,” *Neurocomputing*, vol. 410, pp. 103–113, Oct. 2020, doi: 10.1016/j.neucom.2020.05.047.

[61] V. Boppana and P. Sandhya, “Web crawling based context aware recommender system using optimized deep recurrent neural network,” *J. Big Data*, vol. 8, no. 1, pp. 1–24, Dec. 2021, doi: 10.1186/s40537-021-00534-7.

[62] R. Mahmud, A. N. Toosi, K. Ramamohanarao, and R. Buyya, “Context-Aware Placement of Industry 4.0 Applications in Fog Computing Environments,” *IEEE Trans. Ind. Informatics*, vol. 16, no. 11, pp. 7004–7013, Nov. 2020, doi: 10.1109/TII.2019.2952412.

[63] S. I. Shafiq, C. Sanin, C. Toro, and E. Szczerbicki, “Virtual Engineering Object VEO,” *Cybern. Syst.*, vol. 46, pp. 35–50, Feb. 2015, doi: 10.1080/01969722.2015.1007734.

[64] S. I. Shafiq, C. Sanin, C. Toro, and E. Szczerbicki, “Virtual engineering object (VEO): Toward experience-based design and manufacturing for industry 4.0,” *Cybern. Syst.*, vol. 46, pp. 35–50, 2015, doi: 10.1080/01969722.2015.1007734.

[65] V. Chang, “An overview, examples, and impacts offered by Emerging Services and Analytics in Cloud Computing virtual reality,” *Neural Comput. Appl.*, 2018, doi: 10.1007/s00521-017-3000-1.

[66] A. S. Sohal, R. Sandhu, S. K. Sood, and V. Chang, “A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments,” *Comput. Secur.*, vol. 74, pp. 340–354, May 2018, doi: 10.1016/J.COSE.2017.08.016.

[67] M. Cottrell and M. Verleysen, “Advances in Self-Organizing Maps,” *Neural Networks*, vol. 19, no. 6–7, pp. 721–722, Jul. 2006, doi: 10.1016/J.NEUNET.2006.05.011.

[68] “GitHub - RealImpactAnalytics/trumania: Trumania is a scenario-based random dataset generator library in python 3.” https://github.com/RealImpactAnalytics/trumania (accessed Mar. 07, 2022).

[69] S. Haykin, *Neural Networks and Learning Machines*. 2008.

[70] P. J. Jupyter Community, “Project Jupyter,” *Project Jupyter*, 2018. .

[71] P. Shukla, T. Gupta, A. Saini, P. Singh, and R. Balasubramanian, “A deep learning frame-work for recognizing developmental disorders,” *Proc. - 2017 IEEE Winter Conf.*

Appl. Comput. Vision, WACV 2017, pp. 705–714, May 2017, doi: 10.1109/WACV.2017.84.

[72] N. S. Chandel, S. K. Chakraborty, Y. A. Rajwade, K. Dubey, M. K. Tiwari, and D. Jat, "Identifying crop water stress using deep learning models," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5353–5367, Sep. 2020, doi: 10.1007/S00521-020-05325-4/FIGURES/8.

[73] R. Yaramasu, V. Bandaru, and K. Pnvr, "Pre-season crop type mapping using deep neural networks," *Comput. Electron. Agric.*, vol. 176, p. 105664, Sep. 2020, doi: 10.1016/J.COMPAG.2020.105664.

[74] B. Saravi, A. P. Nejadhashemi, and B. Tang, "Quantitative model of irrigation effect on maize yield by deep neural network," *Neural Comput. Appl.*, vol. 32, no. 14, pp. 10679–10692, Jul. 2020, doi: 10.1007/S00521-019-04601-2/FIGURES/11.

[75] S. Khaki and L. Wang, "Crop Yield Prediction Using Deep Neural Networks," pp. 139–147, Jun. 2019, doi: 10.1007/978-3-030-30967-1_13.

[76] P. Nevavuori, N. Narra, and T. Lipping, "Crop yield prediction with deep convolutional neural networks," *Comput. Electron. Agric.*, vol. 163, p. 104859, Aug. 2019, doi: 10.1016/J.COMPAG.2019.104859.

[77] S. H. Bhojani and N. Bhatt, "Wheat crop yield prediction using new activation functions in neural network," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13941–13951, Sep. 2020, doi: 10.1007/S00521-020-04797-8/FIGURES/7.

[78] T. Khan, J. Qiu, M. A. A. Qureshi, M. S. Iqbal, R. Mehmood, and W. Hussain, "Agricultural Fruit Prediction Using Deep Neural Networks," *Procedia Comput. Sci.*, vol. 174, pp. 72–78, Jan. 2020, doi: 10.1016/J.PROCS.2020.06.058.

[79] S. Hiriyannaiah, S. G M, K. M H M, and K. G. Srinivasa, "A comparative study and analysis of LSTM deep neural networks for heartbeats classification," *Health Technol. (Berl).*, vol. 11, no. 3, pp. 663–671, May 2021, doi: 10.1007/s12553-021-00552-8.

[80] A. Malali, S. Hiriyannaiah, G. M. Siddesh, K. G. Srinivasa, and N. T. Sanjay, "Supervised ECG wave segmentation using convolutional LSTM," *ICT Express*, vol. 6, no. 3, pp. 166–169, Sep. 2020, doi: 10.1016/j.icte.2020.04.004.

[81] Y. Moriyama, H. Ogasawara, E. Yuda, and J. Hayano, "Assessment of Sleep Quality by Electrocardiogram: Usefulness for Risk Stratification among Hemodialysis Patients with End-stage Renal Disease," *2018 IEEE 7th Glob. Conf. Consum. Electron. GCCE 2018*, pp. 408–409, Dec. 2018, doi: 10.1109/GCCE.2018.8574695.

[82] J. Amarasena, N. Indimagedara, S. Wanigasuriya, B. Bandare, K. Pulasinghe, and J.

Tharmaseelan, “IPillow: Sleep Quality Improvement System,” *2019 Int. Conf. Adv. Comput. ICAC 2019*, pp. 427–433, Dec. 2019, doi: 10.1109/ICAC49085.2019.9103396.

[83] J. Moo Estrella and G. Arankosky Sandoval, “Comparison and prediction of sleep quality in users of bed or hammock as sleeping device,” *Sleep Heal.*, vol. 7, no. 1, pp. 93–97, Feb. 2021, doi: 10.1016/J.SLEH.2020.06.002.

[84] K. Slyusarenko and I. Fedorin, “Smart alarm based on sleep stages prediction,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, pp. 4286–4289, Jul. 2020, doi: 10.1109/EMBC44109.2020.9176320.

[85] T. Nakamura, Y. D. Alqurashi, M. J. Morrell, and D. P. Mandic, “Hearables: Automatic Overnight Sleep Monitoring with Standardized In-Ear EEG Sensor,” *IEEE Trans. Biomed. Eng.*, vol. 67, no. 1, pp. 203–212, Jan. 2020, doi: 10.1109/TBME.2019.2911423.

[86] J. M. Kortelainen, M. O. Mendez, A. M. Bianchi, M. Matteucci, and S. Cerutti, “Sleep staging based on signals acquired through bed sensor,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 776–785, May 2010, doi: 10.1109/TITB.2010.2044797.

[87] M. B. Pouyan, J. Birjandtalab, M. Heydarzadeh, M. Nourani, and S. Ostadabbas, “A pressure map dataset for posture and subject analytics,” *2017 IEEE EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2017*, pp. 65–68, Apr. 2017, doi: 10.1109/BHI.2017.7897206.

[88] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, Oct. 2015, doi: 10.1109/COMST.2015.2444095.

[89] R. Sadeghi, T. Banerjee, and J. Hughes, “Predicting Sleep Quality in Osteoporosis Patients Using Electronic Health Records and Heart Rate Variability,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, pp. 5571–5574, Jul. 2020, doi: 10.1109/EMBC44109.2020.9175629.

[90] C. Panagiotou, I. Samaras, J. Gialelis, P. Chondros, and D. Karadimas, “A comparative study between SVM and fuzzy inference system for the automatic prediction of sleep stages and the assessment of sleep quality,” *Proc. 2015 9th Int. Conf. Pervasive Comput. Technol. Heal. PervasiveHealth 2015*, pp. 293–296, Dec. 2015, doi: 10.4108/ICST.PERVASIVEHEALTH.2015.259248.

[91] Y. El-Manzalawy, O. Buxton, and V. Honavar, “Sleep/wake state prediction and sleep

parameter estimation using unsupervised classification via clustering," *Proc. - 2017 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2017*, vol. 2017-January, pp. 718–723, Dec. 2017, doi: 10.1109/BIBM.2017.8217742.

[92] E. E. Sham and D. P. Vidyarthi, "Intelligent admission control manager for fog-integrated cloud: A hybrid machine learning approach," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 10, p. e6687, May 2022, doi: 10.1002/cpe.6687.

[93] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, and M. Ambrosin, "An Architectural Vision for a Data-Centric IoT: Rethinking Things, Trust and Clouds," *Proc. - Int. Conf. Distrib. Comput. Syst.*, pp. 1717–1728, Jul. 2017, doi: 10.1109/ICDCS.2017.243.

[94] "A Pressure Map Dataset for In-bed Posture Classification v1.0.0." https://www.physionet.org/content/pmd/1.0.0/ (accessed May 30, 2022).

[95] M. G. Terzano *et al.*, "Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep," *Sleep Med.*, vol. 2, no. 6, pp. 537–553, Nov. 2001, doi: 10.1016/S1389-9457(01)00149-6.

[96] Physionet, "CAP Sleep Database v1.0.0," 2012. https://physionet.org/content/capslpdb/1.0.0/ (accessed Mar. 09, 2022).

[97] V. Jithesh, M. Justin Sagayaraj, and K. G. Srinivasa, "LSTM recurrent neural networks for high resolution range profile based radar target classification," *3rd IEEE Int. Conf.* , Jul. 2017, doi: 10.1109/CIACT.2017.7977298.

[98] S. Tuli, R. Sandhu, and R. Buyya, "Shared data-aware dynamic resource provisioning and task scheduling for data intensive applications on hybrid clouds using Aneka," *Futur. Gener. Comput. Syst.*, vol. 106, pp. 595–606, May 2020, doi: 10.1016/J.FUTURE.2020.01.038.

[99] J. Kang, M. Hanif, E. Mirza, M. A. Khan, and M. Malik, "Machine learning in primary care: potential to improve public health," *J. Med. Eng. Technol.*, vol. 45, no. 1, pp. 75–80, 2021, doi: 10.1080/03091902.2020.1853839.

[100] P. Shimpi, S. Shah, M. Shroff, and A. Godbole, "An artificial neural network approach for classification of vector-borne diseases," *Proc. Int. Conf. Comput. Methodol. Commun. ICCMC 2017*, vol. 2018-January, pp. 412–415, Feb. 2018, doi: 10.1109/ICCMC.2017.8282721.

[101] M. Chovatiya, A. Dhameliya, J. Deokar, J. Gonsalves, and A. Mathur, "Prediction of dengue using recurrent neural network," *Proc. Int. Conf. Trends Electron. Informatics*,

ICOEI 2019, vol. 2019-April, pp. 926–929, Apr. 2019, doi: 10.1109/ICOEI.2019.8862581.

[102] M. S. S. Faruque, S. Banik, and R. M. Rahman, "A fuzzy neuro approach to identify diarrhea epidemic in Bangladesh," *2014 Int. Conf. Informatics, Electron. Vision, ICIEV 2014*, 2014, doi: 10.1109/ICIEV.2014.6850731.

[103] W. Jia, X. Li, K. Tan, and G. Xie, "Predicting the outbreak of the hand-foot-mouth diseases in China using recurrent neural network," *2019 IEEE Int. Conf. Healthc. Informatics, ICHI 2019*, Jun. 2019, doi: 10.1109/ICHI.2019.8904736.

[104] X. Y. Li, Z. Di Xu, and Z. P. Li, "Wavelet neural network predication for epidemic outbreak," *2009 Chinese Control Decis. Conf. CCDC 2009*, pp. 3638–3648, 2009, doi: 10.1109/CCDC.2009.5192879.

[105] F. Saberian, A. Zamani, M. M. Gooya, P. Hemmati, M. A. Shoorehdeli, and M. Teshnehlab, "Prediction of seasonal influenza epidemics in Tehran using artificial neural networks," *22nd Iran. Conf. Electr. Eng. ICEE 2014*, pp. 1921–1923, Dec. 2014, doi: 10.1109/IRANIANCEE.2014.6999855.

[106] M. Abdel-Basset, V. Chang, and N. A. Nabeeh, "An intelligent framework using disruptive technologies for COVID-19 analysis," *Technol. Forecast. Soc. Change*, vol. 163, p. 120431, Feb. 2021, doi: 10.1016/J.TECHFORE.2020.120431.

[107] M. Akhtar, M. U. G. Kraemer, and L. M. Gardner, "A dynamic neural network model for predicting risk of Zika in real time," *BMC Med.*, vol. 17, no. 1, pp. 1–16, Sep. 2019, doi: 10.1186/S12916-019-1389-3/FIGURES/7.

[108] M. Atencia, E. García-Garaluz, H. de Arazoza, and G. Joya, "Estimation of parameters based on artificial neural networks and threshold of HIV/AIDS epidemic system in Cuba," *Math. Comput. Model.*, vol. 57, no. 11–12, pp. 2971–2983, Jun. 2013, doi: 10.1016/J.MCM.2013.03.007.

[109] Y. Bai and Z. Jin, "Prediction of SARS epidemic by BP neural networks with online prediction strategy," *Chaos, Solitons and Fractals*, vol. 26, no. 2, pp. 559–569, Oct. 2005, doi: 10.1016/J.CHAOS.2005.01.064.

[110] L. Pellis *et al.*, "Eight challenges for network epidemic models," *Epidemics*, vol. 10, pp. 58–62, Mar. 2015, doi: 10.1016/J.EPIDEM.2014.07.003.

[111] H. Xue, Y. Bai, H. Hu, and H. Liang, "Influenza Activity Surveillance Based on Multiple Regression Model and Artificial Neural Network," *IEEE Access*, vol. 6, pp. 563–575, Nov. 2017, doi: 10.1109/ACCESS.2017.2771798.

[112] "Zika Virus Epidemic | Kaggle," 2016. https://www.kaggle.com/cdc/zika-virus-epidemic/code (accessed Mar. 02, 2022).

[113] "WHO EMRO | Infectious diseases | Health topics," 2020. http://www.emro.who.int/health-topics/infectious-diseases/index.html (accessed Mar. 09, 2022).

[114] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[115] P. Wang, X. Zheng, G. Ai, D. Liu, and B. Zhu, "Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in Russia, Peru and Iran," *Chaos, Solitons and Fractals*, vol. 140, p. 110214, Nov. 2020, doi: 10.1016/j.chaos.2020.110214.

[116] A. L. Caterini and D. E. Chang, "Recurrent Neural Networks," *SpringerBriefs Comput. Sci.*, vol. 0, no. 9783319753034, pp. 59–79, 2018, doi: 10.1007/978-3-319-75304-1_5.

[117] X. Zhu *et al.*, "Attention-based recurrent neural network for influenza epidemic prediction," *BMC Bioinformatics*, vol. 20, no. 18, pp. 1–10, Nov. 2019, doi: 10.1186/S12859-019-3131-8/FIGURES/5.

[118] Z. Li, X. Luo, B. Wang, A. L. Bertozzi, and J. Xin, "A Study on Graph-Structured Recurrent Neural Networks and Sparsification with Application to Epidemic Forecasting," *Adv. Intell. Syst. Comput.*, vol. 991, pp. 730–739, Feb. 2019, doi: 10.48550/arxiv.1902.05113.

[119] P. Yomwan, C. Cao, P. Rakwatin, W. Suphamitmongkol, R. Tian, and A. Saokarn, "A study of waterborne diseases during flooding using Radarsat-2 imagery and a back propagation neural network algorithm," *Geomatics, Nat. Hazards Risk*, vol. 6, no. 4, pp. 289–307, May 2015, doi: 10.1080/19475705.2013.853325.

[120] R. Salgotra, M. Gandomi, and A. H. Gandomi, "Time Series Analysis and Forecast of the COVID-19 Pandemic in India using Genetic Programming," *Chaos, Solitons & Fractals*, vol. 138, p. 109945, Sep. 2020, doi: 10.1016/J.CHAOS.2020.109945.

[121] R. Nijhawan, J. Das, and B. Raman, "A hybrid of deep learning and hand-crafted features based approach for snow cover mapping," *Int. J. Remote Sens.*, vol. 40, no. 2, pp. 759–773, Jan. 2019, doi: 10.1080/01431161.2018.1519277.

[122] S. Han, H. S. Choi, J. Choi, J. H. Choi, and J. G. Kim, "A DNN-based data-driven modeling employing coarse sample data for real-time flexible multibody dynamics simulations," *Comput. Methods Appl. Mech. Eng.*, vol. 373, p. 113480, Jan. 2021, doi:

10.1016/J.CMA.2020.113480.

[123] H. Lau *et al.*, “The positive impact of lockdown in Wuhan on containing the COVID-19 outbreak in China,” *J. Travel Med.*, vol. 27, no. 3, pp. 1–7, May 2020, doi: 10.1093/JTM/TAAA037.

[124] A. Arora, P. Chakraborty, M. P. S. Bhatia, and P. Mittal, “Role of Emotion in Excessive Use of Twitter During COVID-19 Imposed Lockdown in India,” *J. Technol. Behav. Sci.*, vol. 6, no. 2, pp. 370–377, Oct. 2021, doi: 10.1007/s41347-020-00174-3.

[125] A. Arora, P. Chakraborty, and M. P. S. Bhatia, “Problematic Use of Digital Technologies and Its Impact on Mental Health During COVID-19 Pandemic: Assessment Using Machine Learning,” *Stud. Syst. Decis. Control*, vol. 348, pp. 197–221, 2021, doi: 10.1007/978-3-030-67716-9_13.

[126] D. Ibarra-Vega, “Lockdown, one, two, none, or smart. Modeling containing covid-19 infection. A conceptual model,” *Sci. Total Environ.*, vol. 730, p. 138917, Aug. 2020, doi: 10.1016/J.SCITOTENV.2020.138917.

[127] J. W. Sonn, M. Kang, and Y. Choi, “Smart city technologies for pandemic control without lockdown,” *https://doi.org/10.1080/12265934.2020.1764207*, vol. 24, no. 2, pp. 149–151, Apr. 2020, doi: 10.1080/12265934.2020.1764207.

[128] A. Ahlawat *et al.*, “Smart surveillance using on cloud machine learning and internet controlled UAVS,” *Adv. Math. Sci. J.*, vol. 9, no. 6, pp. 3789–3794, 2020, doi: 10.37418/amsj.9.6.57.

[129] R. Sujath, J. M. Chatterjee, and A. E. Hassanien, “A machine learning forecasting model for COVID-19 pandemic in India,” *Stoch. Environ. Res. Risk Assess.*, vol. 34, no. 7, pp. 959–972, Jul. 2020, doi: 10.1007/S00477-020-01827-8/TABLES/3.

[130] Y. Zhang, X. Yu, H. G. Sun, G. R. Tick, W. Wei, and B. Jin, “Applicability of time fractional derivative models for simulating the dynamics and mitigation scenarios of COVID-19,” *Chaos, Solitons & Fractals*, vol. 138, p. 109959, Sep. 2020, doi: 10.1016/J.CHAOS.2020.109959.

[131] D. Parbat and M. Chakraborty, “A python based support vector regression model for prediction of COVID19 cases in India,” *Chaos, Solitons & Fractals*, vol. 138, p. 109942, Sep. 2020, doi: 10.1016/J.CHAOS.2020.109942.

[132] X. Zhang, R. Ma, and L. Wang, “Predicting turning point, duration and attack rate of COVID-19 outbreaks in major Western countries,” *Chaos, Solitons & Fractals*, vol. 135, p. 109829, Jun. 2020, doi: 10.1016/J.CHAOS.2020.109829.

[133] S. Tuli, S. Tuli, R. Tuli, and S. S. Gill, “Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing,” *Internet of Things*, vol. 11, p. 100222, Sep. 2020, doi: 10.1016/J.IOT.2020.100222.

[134] P. Melin, J. C. Monica, D. Sanchez, and O. Castillo, “Analysis of Spatial Spread Relationships of Coronavirus (COVID-19) Pandemic in the World using Self Organizing Maps,” *Chaos, Solitons & Fractals*, vol. 138, p. 109917, Sep. 2020, doi: 10.1016/J.CHAOS.2020.109917.

[135] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, “Automated detection of COVID-19 cases using deep neural networks with X-ray images,” *Comput. Biol. Med.*, vol. 121, p. 103792, Jun. 2020, doi: 10.1016/J.COMPBIOMED.2020.103792.

[136] A. Rosebrock, “OpenCV People Counter - PyImageSearch,” 2018. https://pyimagesearch.com/2018/08/13/opencv-people-counter/ (accessed Mar. 02, 2022).

[137] Y. Asano, “Learning OpenCV 3 Computer,” *O’Reilly Media, Inc.*, pp. 32–35, 2012, Accessed: Mar. 02, 2022. [Online]. Available: https://www.oreilly.com/library/view/learning-opencv-3/9781491937983/.

[138] “OpenCV: OpenCV Tutorials,” 2016. https://docs.opencv.org/3.0.0/d9/df8/tutorial_root.html (accessed Mar. 08, 2022).

[139] C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer International Publishing, 2018.

[140] “GitHub - CSSEGISandData/COVID-19: Novel Coronavirus (COVID-19) Cases, provided by JHU CSSE.” https://github.com/CSSEGISandData/COVID-19 (accessed Mar. 02, 2022).

[141] E. Dong, H. Du, and L. Gardner, “An interactive web-based dashboard to track COVID-19 in real time,” *Lancet Infect. Dis.*, vol. 20, no. 5, pp. 533–534, May 2020, doi: 10.1016/S1473-3099(20)30120-1/ATTACHMENT/9DE52FFD-61D4-4C3F-826F-841D978D469D/MMC1.PDF.

[142] G. Schwarz, “Estimating the Dimension of a Model,” *https://doi.org/10.1214/aos/1176344136*, vol. 6, no. 2, pp. 461–464, Mar. 1978, doi: 10.1214/AOS/1176344136.

[143] covid19india, “Coronavirus Outbreak in India,” *Covid19India.Org*, 2020. https://www.covid19india.org/.

[144] F. Chollet, “Keras: The Python Deep Learning library,” *Keras.Io*, 2015.

[145] F. Nelli and F. Nelli, “Deep Learning with TensorFlow,” in *Python Data Analytics*, 2018.

[146] P.-N. Tan, M. Steinbach, and V. Kumar, “Introduction to Data Mining Instructor’s Solution Manual.”

List of Publications

Journal Publications

1. H. K. Gill, V. K. Sehgal, and A. K. Verma, "A context sensitive security framework for Enterprise multimedia placement in fog computing environment," *Multimed. Tools Appl.*, vol. 79, no. 15–16, pp. 10733–10749, 2020.
   [SCI/ Scopus Indexed, Impact Factor- 2.757]
2. H. K. Gill, V. K. Sehgal, and A. K. Verma, "CASE-CF: Context Aware Smart Epidemic Control Framework," *New Gener. Comput.*, no. 0123456789, 2021.
   [SCI/ Scopus Indexed, Impact Factor- 1.113]
3. H. K. Gill, V. K. Sehgal, and A. K. Verma, "A deep neural network based context-aware smart epidemic surveillance in smart cities," *Libr. Hi Tech*, 2021.
   [SCI/ Scopus Indexed, Impact Factor- 2.357]

Papers Presented in Conferences:

1. H. K. Gill, V. K. Sehgal, and A. K. Verma, "Sleep Quality and Best Posture Prediction using Contextual Body Sensors using LSTM," in ***IEEE** 2021 Asian Conference on Innovation in Technology*, 2021, pp. 1–4.
2. H.K. Gill, V. K. Sehgal, and A. K. Verma, "A Context Aware Recommender System for Predicting Crop Factors using LSTM," in ***IEEE** 2021 Asian Conference on Innovation in Technology*, 2021, pp. 1–4.