# Pharmaceutical Demand Forecasting of Time-series data with Recurrent Neural Networks, NeuroEvolution of Augmenting Topologies and Traditional Machine Learning Models: A Comparative Study


**Submitted by**

**Pritu Roy**
ID: 1808002

**Uthso Roy Shuvo**
ID: 1808009

**Submitted to**

**Dr. Sultana Parveen**

Professor

Department of Industrial and Production Engineering

Bangladesh University of Engineering and Technology



**Department of Industrial and Production Engineering**

**Bangladesh University of Engineering and Technology**



24 June, 2024

# CERTIFICATE OF APPROVAL

The thesis titled “Pharmaceutical Demand Forecasting of Time-series Data with Recurrent Neural Networks, NeuroEvolution of Augmenting Topologies and Traditional Machine Learning Models: A Comparative Study”, submitted by Pritu Roy (ID: 1808002) and Uthso Roy Shuvo (ID: 1808009) has been granted up to the mark for accomplishing all the prerequisite necessary for the degree of Bachelor of Science in Industrial and Production Engineering on 24 June 2024.

Approved by,

_____

Dr. Sultana Parveen

Professor

Department of Industrial and Production Engineering

Bangladesh University of Engineering and Technology

# DECLARATION

We do hereby declare that this complete report titled “Pharmaceutical Demand Forecasting of Time-series Data with Recurrent Neural Networks, NeuroEvolution of Augmenting Topologies and Traditional Machine Learning Models: A Comparative Study” has been prepared by us which is a six-credit course for our Bachelor of Science degree at Bangladesh University of Engineering and Technology. We also declare that neither this report nor any part of it has been submitted elsewhere.

The Authors:

_____

Name: Pritu Roy

Student ID: 1808002

_____

Uthso Roy Shuvo

Student ID: 1808009

## Acknowledgment

We want to acknowledge our deep and sincere gratitude to our supervisor, Dr. Sultana Parveen. We would like to thank her for guiding us through the thesis work. She has helped us understand the methodology to carry out this thesis. Also pointed out the major flaws in our work & helped us correct those We are extremely grateful to our friends and family for constantly supporting us. Their support gave us the courage to carry out such successful thesis work. Finally, our sincere thanks to all the people who spent their time, gave attention, and provided us with the right and essential information during this project work.

# Abstract

Accurate demand forecasting is critical in the pharmaceutical industry for efficient inventory management, production planning, and supply chain optimization and for reducing drug expiration. Ensuring the availability of essential medications while minimizing waste and avoiding stockouts and overstocking is paramount. Traditional trend-based methodologies often fall short due to variations in demand influenced by pricing, promotions, and product quality. To improve forecasting accuracy, this study investigates advanced forecasting methodologies such as neural network-based models and traditional machine learning models for pharmaceutical demand forecasting, specifically Recurrent Neural Networks like Long short-term memory (LSTM) and NeuroEvolution of Augmenting Topologies (NEAT). NEAT's evolutionary technique enhances neural network designs by learning from data and evolving models over time. We compare the performance of NEAT and LSTM to traditional machine learning methods such as XGBoost and Random Forest in forecasting pharmaceutical drug demand across various drug categories. The purpose of this study is to develop and validate specialized forecasting methods tailored to the unique demand patterns of different drug classifications, thereby assisting pharmaceutical businesses in managing market volatility, optimizing resources, and meeting healthcare demands. Through comprehensive comparative analysis and evaluation using metrics such as accuracy, mean absolute error and root mean square error, this study aims to construct realistic forecast models that significantly enhance the industry's ability to respond to dynamic market conditions.

# List of Figures

# List of Tables

# Table of Contents

# Chapter 1 : Introduction

## 1.1 Background

In today's intense competition, a company must have a reasonable estimate of how vital indicators will behave in the future. One of the most important key indicators is demand. An efficient forecasting system of demand can improve resource utilization and reduce inventories. Demand forecasting is paramount in the pharmaceutical sector to ensure efficient inventory management, production planning, and optimization of supply chains. With pharmaceuticals often having limited shelf lives and specific storage requirements, accurate forecasting helps prevent stockouts or overstocking, ensuring the availability of critical medicines when needed. It enables companies to plan production schedules effectively, allocate resources efficiently, and minimize waste. Moreover, by understanding demand patterns, pharmaceutical companies can stay competitive, anticipate market trends, and develop targeted marketing strategies. Demand forecasting varies from industry to industry due to the variation in demand patterns and the influence of varying factors. The pharmaceutical industry is one of the kinds that face demand fluctuations. The demand variation occurs in this sector due to price variation, promotional activity, no. of different alternatives, product availability, quality, etc. In such cases, simple trend and seasonality-based forecasting are not appropriate. The incorporation of different factors will provide accurate forecasts in such cases. Accurate and reliable forecasting methods can improve patient care by ensuring timely access to life-saving medications and contribute to overall public health outcomes.

### 1.1.1 Condition of the pharmaceutical industry

The pharmaceutical industry is undergoing significant changes in the context of demand forecasting, driven by factors such as changing and evolving patient demographics, advancements in healthcare technology, and the increasing complexity of drug development and distribution. Traditional forecasting methods, while useful, may struggle to keep pace with these changes due

to their limitations in handling large and complex datasets, identifying non-linear relationships, and adapting to dynamic market conditions.

### 1.1.2 Currently employed forecasting approaches

A good forecasting model leads to improved customer satisfaction, reduced cost, increased sales revenue, and efficient production plans [1]. The methodologies that have been used in sales forecasting are typically time series algorithms that can be classified as linear or nonlinear, depending on the nature of the model they are based on [2]. There are currently at least 70 time series forecasting models using linear and, or nonlinear structures for quantitative demand forecasting [3]. Many theoretical and heuristic models have been developed and evaluated empirically in recent decades [4]. In general, linear mathematical models are preferred over more complex models, as linear models have the important practical advantage of accessible interpretation and implementation [5]. On the other hand, if linear models fail to perform well in both sample fitting and out-of-sample forecasting, more complex nonlinear models should be considered [6]. The most common and useful mathematical model used to predict demand is ANN. It has been successfully applied in sales or demand forecasting. Kue et al. used this tool to forecast sales for a beverage company and found that this approach is better than ARIMA [7]. Chang et al. applied a fuzzy BPN to forecast sales for the Taiwanese printed circuit board industry [8]. Hyunchul et al. first used independent component analysis to screen variables before employing an ANN algorithm to predict sales for a Korean shopping mall. They also showed that the proposed forecasting scheme is superior to a forecasting method in which principal component analysis is first used to screen variables before an ANN algorithm is applied [9]. In addition, the ANN is compared with the Box-Jenkins models using international airline passenger traffic, domestic car sales, and foreign car sales in the USA. It was concluded that the Box-Jenkins models outperformed the ANN models in short-term forecasting, although the ANN models outperformed the Box-Jenkins for long-term forecasting [10]. Zhang did a comprehensive review of the literature concerning the utilization of ANNs in forecasting problems in various areas. This method performed equally well with linear methods in 30% and better in 56% of the cases reviewed. In a subsequent study by Stock et al. linear and nonlinear methods were compared and it was found that in terms of forecasting performance, combinations of nonlinear methods are better than combinations of linear methods. Additionally, feed-forward neural networks (FNNs) that constitute a distinctive ANN architecture performed equally well or better than traditional methods in more than half of the cases

[11]. Overall, it can be inferred that, among other methods, ANN has been successfully used in forecasting.

## 1.2 The rationale of the study

Pharmaceutical companies manage diverse product lines and face market volatility influenced by regular changes, patent expirations, and competitive pressures, So, accurate demand forecasting becomes paramount here. Advanced forecasting methods offer the capability to analyze vast and diverse datasets, enabling companies to anticipate market trends, optimize inventory management, and tailor marketing strategies to specific customer segments. Using advanced techniques such as machine learning, neural networks, and predictive analytics, pharmaceutical companies can gain a competitive edge, improve operational efficiency, ensure regulatory compliance, and ultimately enhance their ability to meet the evolving needs of healthcare providers and patients. Neural networks offer the advantage of learning from data and adjusting their models over time, which leads to improved forecasting accuracy compared to traditional statistical methods. Additionally, they can adapt to changing conditions and incorporate new information, enhancing the reliability of predictions. Here to increase the predicting efficiency of Neural Networks, we intend to use NeuroEvolution Augmented Topologies (NEAT) which uses a principled method of crossover of different topologies and incrementally grows from a minimal structure ensuring fast neural network learning. We will compare the NEAT model with different machine learning algorithms such as: XGBoost, Random Forest and analysis the performances of each model. The scopes of the study are mentioned below:

- This study is applicable to AI based decision-making regarding the forecasting of necessary drugs which will help to reduce to dependency on traditional statistical method of forecasting.
- This study will also help them in decision-making regarding resource utilization and inventory planning by developing a forecast model to predict the uncertain demand of Pharmaceutical Drugs.

## 1.3 The objective of the study

Accurate demand forecasting is critical for successful strategic planning and operational efficiency in industries including pharmaceutical companies. The choice of a specific forecasting model depends on factors such as the characteristics of the time series data and the desired level of complexity. Considering the situation, the objective of the study has been set. The objectives of the study are mentioned below:

1. To compare the performance of neural network-based models and traditional machine learning algorithms in forecasting pharmaceutical drug demand across various drug categories.
2. To develop and validate specialized forecasting methods tailored to the unique demand patterns of different drug classifications.
3. Investigating the Potential of NEAT for discovering the optimal Neural Network Architecture

## 1.4 Structure of the thesis paper

The thesis outline follows: Chapter One gives the background for selecting the topic and sector, offers insights into previous forecasting work, and states the study's rationale and main objectives. Chapter Two presents a detailed literature review on forecasting, including techniques, Artificial Neural Networks (ANN), NeuroEvolution Augmented Topologies (NEAT), and other learning algorithms. Chapter Three explains the study's methodology. Chapter Four covers data collection, sorting, preparation, and the development of the forecasting model using NEAT-based Neural Networks. Chapter Five discusses the findings and results from the forecasting model. Chapter Six provides the conclusion [12].

# Chapter 2 : Literature Review

## 2.1 Introduction

This chapter presents an in-depth literature review related to the work, covering topics such as different forecasting methods, demand forecasting in pharmaceutical sectors, artificial neural networks (ANN), NeuroEvolution Augmented Topologies (NEAT) for generating Neural Networks, and detailed literature on them.

## 2.2 Forecasting

Forecasting is the technique of predicting future outcomes for given historical entities. Forecasting involves estimating or calculating future occurrences or developments based on a conceptual model. The conceptual model can be a heuristic model formed by individuals' life experiences and social, cultural, and economic circumstances, such as their expectations for daily routines. On the other end, the conceptual model might be analytic, formal, and complicated, using numerical measurements, equations, simulations, and projections.

### 2.2.1 Types of forecasting methods

Forecasting approaches are separated into two types: quantitative and qualitative. Quantitative approaches involve mathematical models including moving averages, straight-line projections, exponential smoothing, regression, analysis, decomposition, Box-Jenkins, expert systems, and neural networks. Qualitative models include subjective or intuitive models like jury or management opinion, sales force composite, and customer expectations [13, 14]. Some forecasting approaches are briefly covered here [13,14]:

- **Quantitative Forecasting**:
  - **Regression Analysis**: This method statistically correlates sales with independent variables like price fluctuations, competition, and economics.

- **Exponential smoothing:** This calculates a weighted average based on prior sales, trends, and seasonality.
- **Moving average:** This method calculates the average of a set number of observations.
- **Box-Jenkins:** From past sales data and forecasting errors and autoregressive moving average forecasts are developed using the autocorrelated structure.
- **Trend line analysis:** This method fits a line to the sales data by minimizing the squared error between the line and actual past sales data.
- **Decomposition:** This method divides the sales data into seasonal, cyclical, trend, and noise components and projects them in the forecast.
- **Straight-line projection** is a graphic extrapolation of the previous data, which is projected the forecast into the forecast to calculate future data.
- **Life-cycle analysis** bases the forecast upon the life cycle of products, which the introduction, growth, maturity, or decline stage of the life cycle.
- **Simulation:** The simulation model is a mathematical model of the actual entity.
- **Expert systems:** In this method, decision rules are developed using the knowledge of one or more experts to derive a forecast.
- **Neural networks** look for patterns in the previous history of sales and analytical data to reveal connections. These connections are used to produce the forecast.

➢ **Qualitative Forecasting Methods**

- **The jury of executive opinion** combines prominent executives' perspectives on forthcoming events.
- **Customer expectations:** Salespeople gather customer expectations and forecast accordingly.
- **Delphi model:** This strategy leverages expert knowledge while ensuring participant confidentiality.
- **Salesforce:** This method individually combines the forecast of sales personnel.

### 2.2.2 Application of forecasting in different areas

Forecasting is widely utilized for several purposes, including short-term or long-term gas and electricity load forecasting, tourist flow estimation, and demand forecasting for industrial, food, and clothing items. According to a study, the fashion business offers unique opportunities for sales forecasting. Forecasting is tough due to the long time-to-market and short product life cycles [15]. Forecasting accuracy is even more important in the business operations of the foodstuff retail segment because retailing is well-known as a competitive industry in both established and developing economies [16-18]. Load forecasting aids in gas supply planning, maintenance, and distribution dispatching across the network. Accurate prediction can increase operating efficiency, conserve energy, and lower expenses. Thus, natural gas consumption forecasting has been extensively explored [19-21]. Wind speed forecasting is crucial for wind farm management and safe integration into the electric power grid [22]. Here are some instances of forecasting applications.

## 2.3 Machine Learning for Forecasting

Selecting the right forecasting technique for a given situation gets harder as the number of parameters and techniques rises. Using a class of algorithms known as "universal approximators," which can approximate any function to any degree of accuracy, is one potential remedy. Any necessary function between historical and future data can be learned using such universal approximators, thereby reducing alternative forecasting methods to a subset of the functions that the universal approximators can learn. Artificial neural networks and support vector machines, two examples of machine learning approaches, are universal approximators that can be used to learn any function. Supply chain time series, for example, have a very noisy data domain that must be forecasted. It is highly desirable to disregard the noise and only identify actual patterns in the data that will recur in the future. The two key characteristics of ML-based methods that make them effective for supply chain forecasting issues in the face of noise are (i) their capacity to learn any function and (ii) their capacity to regulate the learning process [35].

## 2.4 Artificial Neural Network (ANN)

Artificial neural networks (ANNs) connect mathematical nodes or neurons to model complicated functional interactions [23]. It was developed in the 1940s to assist cognitive scientists in comprehending the complexities of the neural system. It has evolved steadily and has been accepted in a wide range of scientific disciplines. ANNs are numerical structures that draw inspiration from the human brain's learning process. These mathematical methods can be used to handle a variety of problems, including system identification, forecasting, pattern recognition, classification, and process control [24].

### 2.4.1 ANN Architecture

The architecture has three types of neuron layers: input, hidden, and output layers. In a feed-forward network, the signal flows from the input to the output unit in a specific direction. Although data can be processed across many layers, there are no feedback linkages. Recurrent systems use feedback connections. Unlike a feed-forward network, the network's dynamic features are very crucial. As the network reaches a steady state, the activation levels will no longer change. In some applications, changes in output neuron activation values are significant due to the network's dynamic character [25].

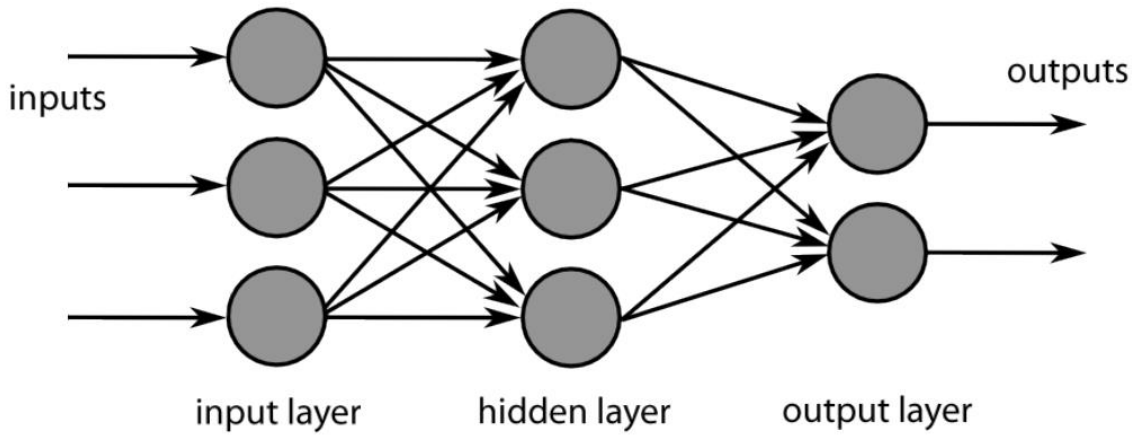


**Figure 1: Neural Network [26]**

A neural network needs to be set to process inputs and produce outputs. There are various methods available to determine the strength of connections. One approach is to use existing information to set the network's weights. Another way to train the neural network is by feeding the training pattern and altering the weights based on a learning rule [27].

### 2.4.2 ANN Learning

During the learning of an artificial neural network, the external world interacts with it. During this phase, a learning rule adjusts the network's adjustable parameters and matrix elements. In this sense, "learning" and "adaptation" are seen as changes in network parameters. The external world often provides "training" input vectors in ANN models. There are three types of learning: supervised, unsupervised, and reinforcement. Supervised learning involves presenting an input vector and a set of anticipated responses to each node at the output layer, one for each node, at the output layer. Supervised learning applications include the backpropagation method, delta rule, and perceptron rule. Unsupervised learning involves training an output layer to identify clusters of patterns in input data. This paradigm aims to identify statistically significant features in the input layer. In contrast to supervised learning, unsupervised learning requires the system to create its representation of the input layer rather than a predetermined set of categories. Reinforcement learning does not assume prior knowledge. Learners must experiment to determine which steps are most rewarding. In fascinating and hard circumstances, efforts can impact the immediate premium and the subsequent state through all subsequent rewards. These two characteristics, trial-and-error search, and delayed reward, are the two most important distinguishing features of reinforcement learning [28].

### 2.4.3 Training and Testing Neural Networks

For the best training process, it's important to collect diverse examples that demonstrate all aspects of the problem. Typically, the network is trained for a predetermined number of periods or when the output error falls below a specific threshold. Avoid overtraining the network. Overtraining might cause the network to become overly adapted to the training set, hindering its ability to reliably identify samples outside of that set [28].

## 2.5 NeuroEvolution of Augmenting Topologies (NEAT)

NEAT is an evolutionary algorithm that creates artificial neural networks [29]. NEAT is considered a form of reinforcement learning because it optimizes neural networks based on performance feedback derived from interacting with an environment. Neuroevolution (NE), the artificial evolution of neural networks using genetic algorithms, has shown great promise in reinforcement learning tasks. NEAT incrementally elaborates structure in a stochastic manner from a minimal starting point. Smaller structures optimize faster, so NEAT can find solutions faster than other neuroevolution methods [30]. To improve the performance of neural networks, there are four aspects to work on:

**a) Genetic Encoding**

Each genome contains a list of connection genes, each indicating a relationship between two node genes. Each connection gene defines the in-node, out-node, and weight of the connection. Mutations in NEAT can alter both connection weights and network architecture. In a NE system, connection weights change depending on whether each link is perturbed. NEAT can create networks with any number of inputs and outputs. Mutations in NEAT can alter both connection weights and network architecture. Mutation creates genomes of varied sizes, often with unique connections at the same places [30].

**b) Tracking Genes through Historical Markings**

To execute crossover, the system must identify genes that match across individuals in the population. Two genes with the same historical origin share the same structure, albeit with differing weights, as they were derived from the same ancestral gene at some point in the past. To determine whether genes align with each other, a system only needs to trace their historical origin [30].

### c) Protecting Innovation through Speciation

Adding new structure to a network can initially impair its fitness. NEAT causes people to compete primarily within their specific niches, rather than the population as a whole. This protects topological inventions and allows them to optimize their structures before competing with other niches. Speciation is often utilized in optimizing multimodal functions and coevolving modular systems to preserve diversity [31].

### d) Minimizing Dimensionality

NEAT starts with a uniform population of networks that have no hidden nodes. NEAT protects innovation through speciation, allowing for a modest start and gradual growth of new structures. Structural mutations add new structures incrementally, and only useful ones survive through fitness evaluations. NEAT searches through fewer weight dimensions, minimizing the number of generations required to discover a solution [30].

NEAT optimizes architecture by reducing the search space dimensionality of connection weights. Growing topologies incrementally from their minimum counterparts results in the ideal topology with the least dimensional space, leading to significant advances in training speed and accuracy. Dynamic topologies and parameters were applied to automate the creation of neural network architecture. This could reduce human effort and improve network effectiveness [32].

### 2.5.1 Application of NEAT in different areas

NEAT is commonly utilized in Behavioral Exploration with Novelty Search. Traditional reinforcement learning systems reward the agent for progress toward an objective computed using an objective function. Such reward functions are frequently misleading and do not promote investigation. Gomes et al. [33] used novelty search in conjunction with NEAT to optimize neural controller development for homogeneous robot swarms. They discovered that novelty search is unaffected by deceit and is particularly effective for bootstrapping evolution. It may identify answers that are less complex than fitness-based evolution while also assisting in the discovery of a wide variety of solutions to the same problem. FS-NEAT is used for automated feature selection. Feature Selection (FS) selects features that contribute significantly to the target problem's outcome. FS-NEAT (Feature Selective NEAT) [24] automates this procedure by using the NEAT algorithm's foundations. FS-NEAT starts with a population of networks with only one input-output link, unlike NEAT, which connects all inputs and outputs. This approach optimizes feature selection by eliminating unnecessary links in the final population [32].

## 2.6 Long Short-Term Memory (LSTM)

Deep learning neural networks have shown promising results in non-linear sequence learning. Deep learning is a new approach to machine learning that employs deep neural networks to create intelligent models. RNNs and LSTM networks are widely used deep learning approaches that outperform traditional machine learning methods for time-series forecasting [33]. RNN and LSTM networks, contrary to other neural networks, have the property of retaining the information across time 6 steps. Furthermore, improved versions of LSTM networks such as a full gradient version made LSTM networks a suitable choice for non-linear sequence forecasting because it overcomes the problem of vanishing gradient. This updating of LSTM networks made it possible to retain information across long-term time steps which enabled LSTM networks to be used for sequence learning LSTM [34]. LSTM is widely used in time-series prediction, natural language processing, sentiment analysis, image and video captioning.

## 2.7 Traditional Machine Learning Algorithms

Traditional machine learning algorithms refer to a set of techniques and models that have been widely used before the advent of deep learning and neural networks. These algorithms are typically based on statistical principles and are effective for a wide range of tasks, including classification, regression, clustering, and dimensionality reduction. Traditional machine learning methods are the cornerstone of many predictive modeling jobs and remain popular due to their interpretability, efficiency with smaller datasets, and efficacy across a wide range of applications. While deep learning has grown in popularity for jobs involving huge datasets and complicated patterns, classical algorithms remain essential tools in the machine learning toolkit, especially in cases where transparency and interpretability are critical.

### 2.7.1 Random Forest

Random vector growth into "trees" in a random forest classification, with each tree growing without full pruning. Additionally, only a small number of randomly chosen variables are present at each node during the spanning tree phase. Specifically, in the application of randomization to data (rows) and a variable (column). The term "random forest" refers to the process of randomly generating a large number of trees for use in regression analysis and categorization. Each tree in the forest is dependent upon a different random vector, and the vectors are all independently and uniformly distributed. The random forest categorization of the choice with the most votes serves as the basis for the final decision tree, which is based on the random vector potential tree "vote" that is formed [35].

### 2.7.2 XGBoost

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that has gained significant popularity and success in demand forecasting tasks. It is a gradient-boosting implementation that linearly combines weak classifiers to create a strong classifier. XGBoost uses CART and linear classifiers as basic classifiers and conducts second-order Taylor expansion on the cost function to represent extra information. XGBoost allows for parallel split point selection, resulting in faster model training times. The primary idea behind XGBoost is to continuously add weak trees with varying weights to the set. It is widely used in various data mining scenarios and algorithm competitions. The algorithm takes its main advantages in precision, flexibility, and automatic processing of missing values [36]. It is highly effective in demand forecasting due to its ability to handle complex relationships, scalability, robustness to outliers, and flexibility in modeling objectives.

# Chapter 3 : Methodology

## 3.1 Introduction

This chapter outlines the research approach used to conduct this study. The flow chart below summarizes the steps of the study.

## 3.2 Machine Learning Workflow

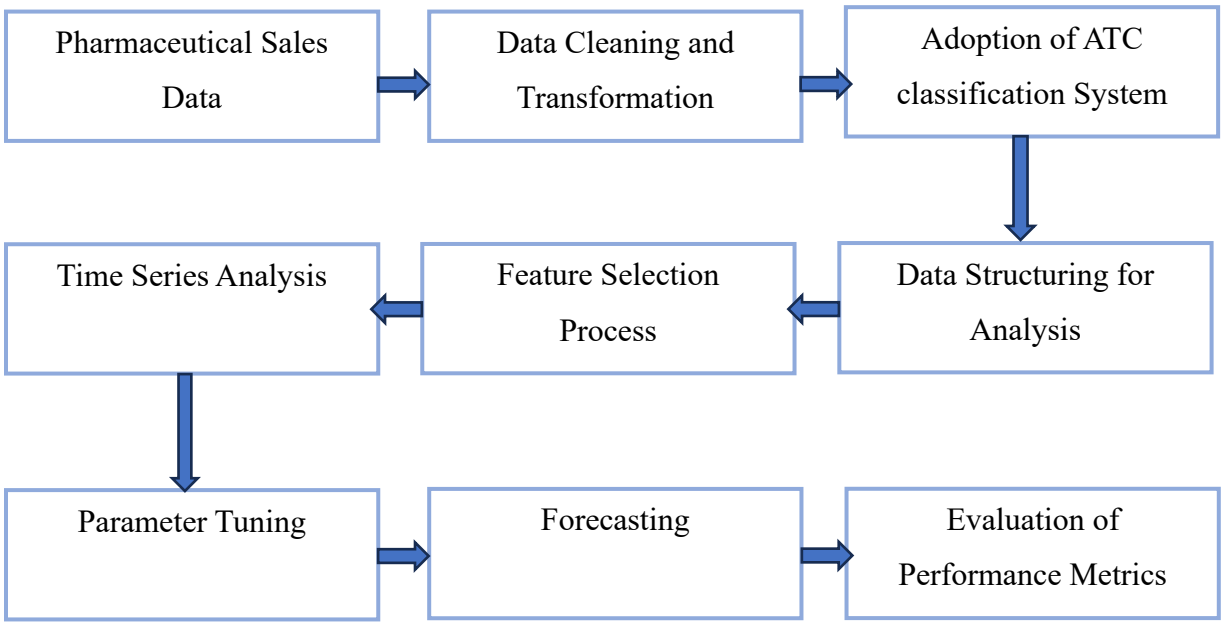


**Figure 2: Generic machine learning workflow**

## 3.3 Algorithm for Neural Network-based Prediction Model

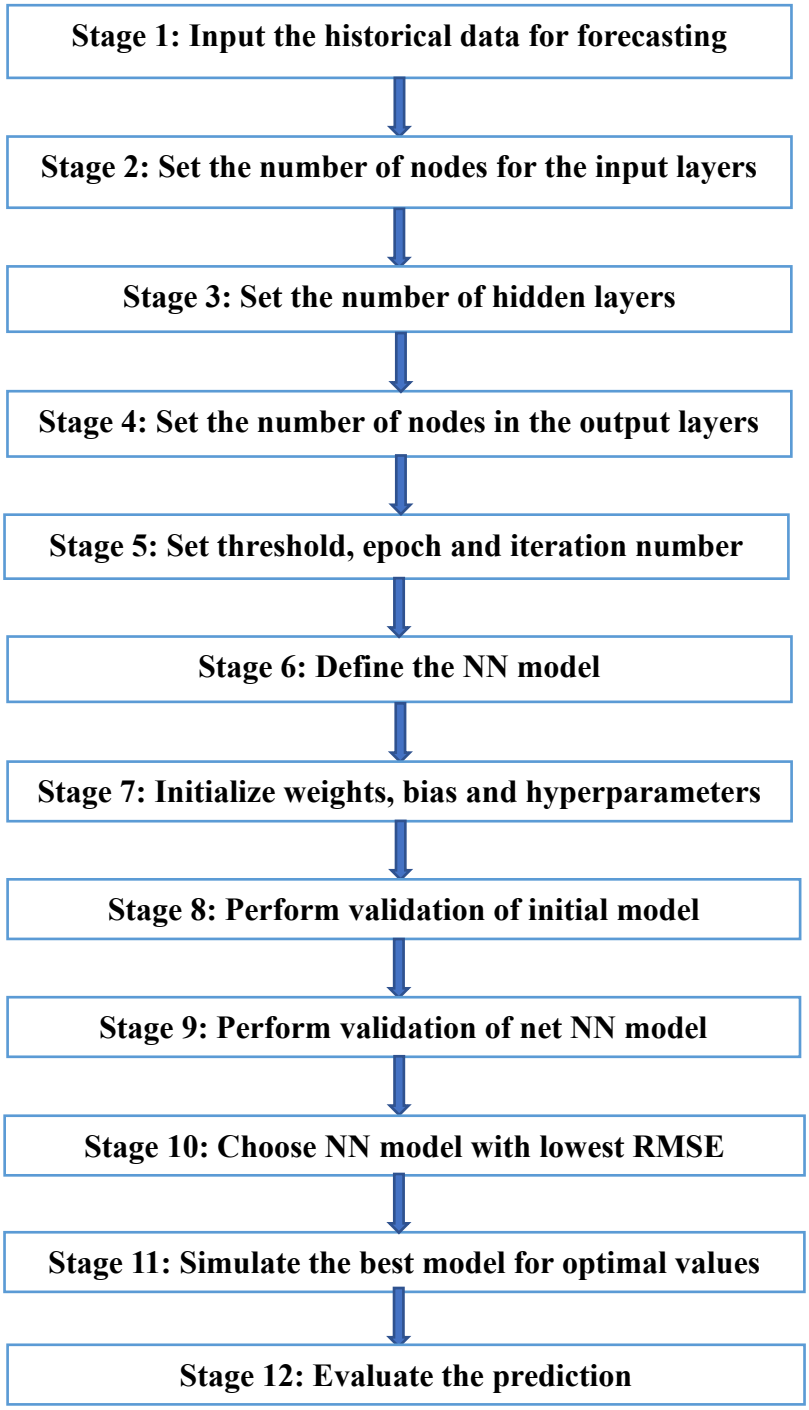


**Figure 3: Neural Network based Prediction Model for Demand Forecasting**

## 3.4 NeuroEvolution Augmenting Topologies

NEAT is a sophisticated algorithm that goes beyond typical neuroevolution techniques by changing both the weights and the topology of neural networks. This makes it especially useful for difficult jobs in which the appropriate network topology is unknown in advance. By using speciation and historical markers, NEAT preserves variety and innovation, resulting in more robust and capable neural network models.

### 3.4.1 Algorithm Overview

NEAT algorithm is an evolutionary technique used to evolve artificial neural networks (ANNs) by optimizing both their weights and topologies.

**Genomes and Phenotypes:** It involves genomes and phenotypes to evolve neural networks

- **Genomes** encode neural networks with nodes (neurons) and connections (synapses).
- **Phenotypes** are the actual ANNs constructed from these genomes.

**Crossover and Mutation:**

- **Crossover** combines genomes from two-parent networks to produce offspring, preserving innovative traits.
- **Mutation** introduces new features by adjusting existing connection weights, creating new connections between nodes, and adding nodes by splitting existing connections.

**Tracking Genes through Historical Markings:**

The historical markers provide NEAT with a significant additional capability. Figure 4 illustrates how the algorithm now recognizes which genes correspond to which. When crossing over, genes with identical innovation numbers are aligned. These genes are called matching genes. Genes that do not match are classified as disjunct or excess, depending on whether they fall within or outside the range of the other parent's innovation numbers. They represent a structure that does not exist in the other genome. When creating kids, genes are randomly selected from either parent with

matching genes, while excess or disjunct genes are always added from the more fit parent. Historical marks enable NEAT to accomplish crossover with linear genomes. In Figure 3, although Parent 1 and Parent 2 appear distinct, their innovation numbers indicate which genes correspond to which. Even without topological analysis, it is possible to form a new structure by combining the overlapping and differing parts of the two parents. Disjoint and surplus genes are inherited from the more fit parent while matching genes are inherited at random. Assuming equal fitness, the disjunct and surplus genes are inherited at random. Disabled genes may reactivate in subsequent generations, as there is a predetermined risk that an inherited gene may be sometimes disabled [30].

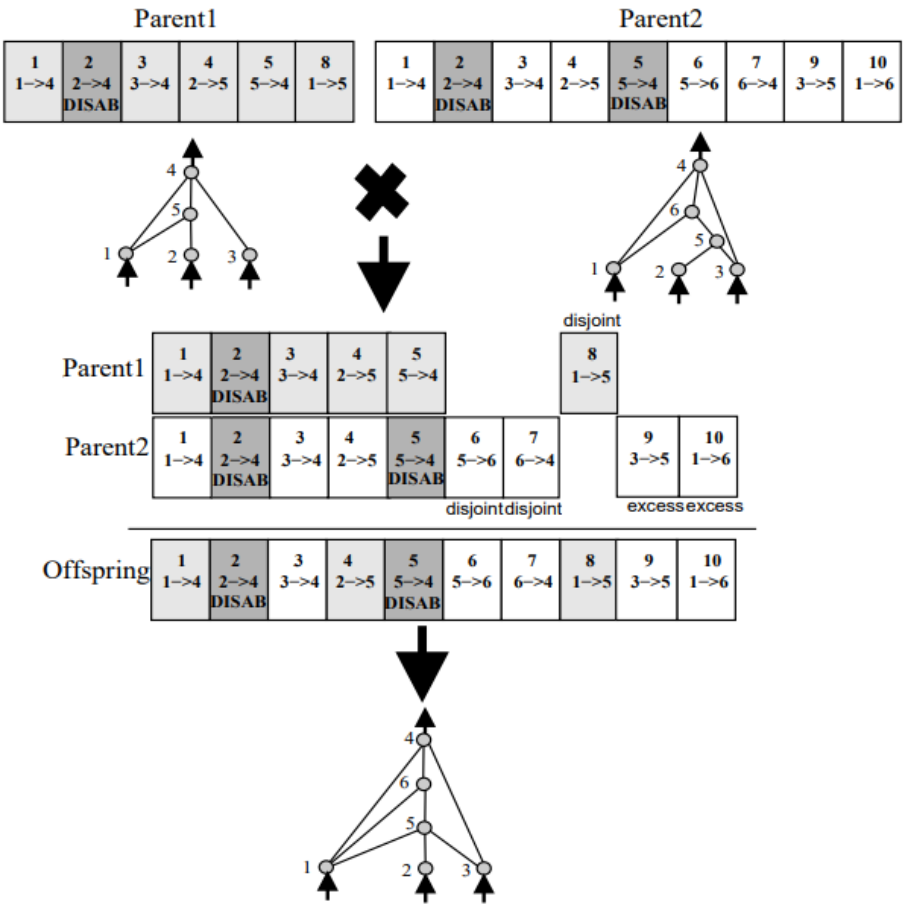


**Figure 4: Matching up genomes for different network topologies [30].**

**Protecting Innovation through Speciation:** Speciating the population allows organisms to compete primarily within their niches instead of with the population. The goal is to separate populations into species with comparable topologies. The challenge appears to be a topology-matching problem. Historical markers are an efficient solution.
The number of excess and disjunct genes between two genomes indicates their compatibility gap. Disjointed genomes share less evolutionary history, making them less compatible. The compatibility distance (δ) between structures in NEAT can be calculated by combining the number of excess E and disjunct D genes with the average weight differences of matching genes (W), including disabled genes [30].

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 . \overline{W} \qquad (1)$$

The coefficients $c_1$, $c_2$, and $c_3$ allow us to adjust the importance of the three factors, and the factor N, the number of genes in the larger genome, normalizes for genome size.

As the reproduction mechanism for NEAT, we use explicit fitness sharing [37], where organisms in the same species must share the fitness of their niche. Thus, a species cannot afford to become too big even if many of its organisms perform well. Therefore, any one species is unlikely to take over the entire population, which is crucial for speciated evolution to work. The adjusted fitness $f_i'$ organism i is calculated according to its distance δ from every other organism j in the population [30]:

$$f_i' = \frac{f_i}{\sum_{j=1}^{n} sh(\delta(i, j))} \qquad (2)$$

## 3.5 Long Short-Term Memory (LSTM)

The LSTM model is a powerful recurrent neural system specially designed to overcome the exploding/vanishing gradient problems that typically arise when learning long-term dependencies, even when the minimal time lags are very long [37].

### 3.5.1 Algorithm Overview

The LSTM model described uses full gradient training to adjust the learnable parameters (weights) involved in the network. More specifically, Backpropagation Through Time is used to compute the weights that connect the different components in the network. Therefore, during the backward pass, the cell state $c^{(t)}$ receives gradients from $y^{(t)}$ as well as the next cell state $c^{(t+1)}$. Those gradients are accumulated before being backpropagated to the current layer [38].

In the last iteration T, the change $\delta_y{}^{(t)}$ corresponds with the network error $\partial E/y^{(T)}$ such that E denotes the loss function. Otherwise, $\delta_y{}^{(t)}$ is the vector of delta values passed down $\Delta^{(t)}$ from the above layer including the recurrent dependencies. This can be done as follows:

$$\delta_y{}^{(t)} = \Delta^{(t)} + R_z{}^T \delta_z{}^{(t+1)} + R_i{}^T \delta_i{}^{(t+1)} + R_f{}^T \delta_f{}^{(t+1)} + R_o{}^T \delta_o{}^{(t+1)} \tag{3}$$

In the second step, the change in the parameters associated with the gates and the memory cell are calculated as:

$$\delta_{0^t} = \delta_y{}^{(t)} \odot h(c^{(t)}) \odot \sigma'(\hat{o}^{(t)}) \tag{4}$$

$$\delta_c{}^{(t)} = \delta_y{}^{(t)} \odot o^{(t)} \odot h'(c^{(t)}) + p_0 \odot \delta_o{}^{(t)} + p_i \odot \delta_o{}^{(t+1)} + p_f \odot \delta_f{}^{(t+1)} + \delta_c{}^{(t+1)} \odot f^{(t+1)} \tag{5}$$

$$\begin{aligned}\delta_{f(t)} &= \delta_{c(t)} \odot c^{(t-1)} \odot \sigma'(\hat{f}^{(t)}) \\ \delta_{i(t)} &= \delta_{c(t)} \odot z^{(t)} \odot \sigma'(\hat{i}^{(t)}) \\ \delta_{z(t)} &= \delta_{c(t)} \odot i^{(t)} \odot g'(\hat{z}^{(t)})\end{aligned} \tag{6}$$

where $\hat{o}^{(t)}$, $\hat{i}^{(t)}$, $\hat{z}^{(t)}$ and $\hat{f}^{(t)}$ denote the raw values attached with the output gate, input gate, block input and forget gate, respectively, before being transformed by the corresponding transfer function.

The delta values for the inputs are only required if there is a layer below that needs to be trained, thus:

$$\delta_x^{(t)} = W_z{}^T \delta_z^{(t)} + W_i^T \delta_i^{(t)} + W_f{}^T \delta_f{}^{(t)} + W_o{}^T \delta_o{}^{(t)} \tag{7}$$

Finally, the gradients for the weights are calculated as follows:

$$\delta_{W*} = \sum_{t=0}^{T} \delta_*^{(t)} \otimes x^{(t)} \qquad \delta_{pi} = \sum_{t=0}^{T-1} c^{(t)} \odot \delta_i^{(t+1)}$$

$$\delta_{R*} = \sum_{t=0}^{T-1} \delta_*^{(t+1)} \otimes y^{(t)} \qquad \delta_{pf} = \sum_{t=0}^{T-1} c^{(t)} \odot \delta_f{}^{(t+1)}$$

$$\delta_{b*} = \sum_{t=0}^{T} \delta_*^{(t)} \qquad \delta_{po} = \sum_{t=0}^{T} c^{(t)} \odot \delta_o{}^{(t)}$$

such that $\otimes$ represents the outer product of two vectors, where $*$ can be any component associated with the weights: the block input $\hat{z}$, the input gate $\hat{i}$, the forget gate $\hat{f}$ or the output gate oˆ.

## 3.6 Random Forest

The random forest algorithm proposed is proposed by Leo Breiman and Adele Cutler, which combines the "Bootstrap aggregating" method and "random subspace method" method to build a set of decision trees, and through the decision tree set to classify. Random forest contains multiple decision tree classifiers, and the output categories are determined by the mode of decision tree classification results. In the construction of a single decision tree, the random forest algorithm uses two random selection processes: the first is the random selection of training samples, and the second is the random selection of the characteristic attributes of the sample. After all the decision trees are constructed, the final classification result is decided by the method of equal-weight voting [39].

### 3.6.1 Algorithm overview

Random forest is an ensemble classifier, which constructs a group of independent and non-identical decision trees based on the idea of randomization. Random forest can be defined as the set $\{h(x, \theta_k), k=1\ldots., L\}$ in which $\theta_k$ is a kind of mutually independent random vector parameter, and x is the input data [40]. Each decision tree uses a random vector as a parameter, randomly selects the feature of samples, and randomly selects the subset of the sample data set as the training set. The construction algorithm of random forest is as follows. k suggests the number of decision trees in the random forest, n indicates the number of samples in the training data set that each decision tree corresponds to, M refers to the feature number of samples, m represents the number of features when carrying out segmentation on a single node of a decision tree, $m<<M$ [41].

## 3.7 XGBoost (Extreme Gradient Boosting)

XGBoost is an optimized, distributed gradient-boosting library. It builds decision tree models sequentially, where each tree attempts to correct errors made by the previous ones. The algorithm introduces regularization to prevent overfitting, handles missing data efficiently, and is designed to work well with sparse data. XGBoost supports parallel processing to enhance training speed and includes features like early stopping, cross-validation, and custom objective functions.

### 3.7.1 Algorithm overview

The main idea of XGBoost is to continuously append weak trees with different weights to the set. The trees in the set have to approach the residuals of the previous prediction as much as possible, which is expressed as follows:

$$\hat{y}_i = \sum_{k=1}^{k} f_k(x_i(x_i) \qquad f_k \in F \tag{8}$$

where $\hat{y}$ is the predicted value, F is the set including all regression trees, $f_k$ is one of the regression trees, and K is the number of regression trees. The predicted value $\hat{y}_i^{(t)}$ is expected to be as close to the true value $y_i$ as possible, and meanwhile without losing its generalization ability. The formula to compute Obj is below

$$Obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_t) + cons\tan t \tag{9}$$

where $l(y_i, \hat{y}_i^{(t)})$ is the loss function, which represents the difference between the predicted value and the true value. It can be any form of loss function which is second-order derivable. $\Omega(f_t)$ is the regularization term, which defines the complexity of the model. The smaller the value of $\Omega(f_t)$ the lower the complexity and the stronger the generalization ability.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|\omega\|^2 \quad (10)$$

Here T is the number of leaf nodes, and $\omega$ is the score represented by the leaf nodes. In terms of operating speed, XGBoost supports the parallel selection of split points, and the model training cost much less time.

## 3.8 Evaluating the Accuracy of the Developed Model

When a forecasting method is in use, one wants to know how well the method predicts what it is supposed to predict. Accuracy is the most common reason to use a specific type of forecast error. The second reason to use a specific forecast error is that it is relatively easy 26 to interpret the error measure. To evaluate the forecast accuracy of the model's statistics such as different statistics are used. They are described below:

### 3.8.1 Performance Metrics

Performance metrics are quantitative measures used to assess the performance of a model, algorithm, or system. They provide insights into how well a model is performing and are essential for comparing different models and selecting the best one. These metrics vary depending on the type of problem being addressed, such as classification, regression, clustering, or forecasting. Below are some common performance metrics used for our models:

**a. Mean Squared Error (MSE)**

Mean Squared Error (MSE) is a standard measure for forecasting errors and its variance is Mean Squared Error (MSE). MSE is related to the standard deviation of forecast errors and is, therefore, an appropriate error for mathematical operations. The mean forecast error deviates from zero and therefore is the mean error present which can be compared to the standard deviation. If instead, the mean error is assumed to be equal to zero the term for mean forecast error is deleted. However, due to the squared function, MSE is more sensitive to outliers and errors smaller than one.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(e_t - \bar{e}_t)^2 \qquad (11)$$

**b. Root Mean Squared Error (RMSE)**

It measures the average magnitude of the errors between predicted values and actual values, providing an overall indication of the accuracy of the model. The RMSE is particularly useful because it penalizes larger errors more heavily than smaller ones, making it sensitive to outliers. The square root of MSE provides an error metric in the same units as the original data.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_t - \bar{e}_t)^2} \qquad (12)$$

**c. Mean Absolute Percentage Error**

It indicates how significant the forecast errors are in comparison to the actual values of the series. Especially useful when the et values are large and can be used to compare the accuracy of the same or different methods on two different time series data.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|e_t - \bar{e}_t\right|}{e_t} \qquad (13)$$

# Chapter 4 : Data Collection & Model Development

## 4.1 Introduction

In the steps of data and information collection, primary data is gathered through interviews and questionnaires, while secondary data comes from literature studies, newspaper articles, and the Internet. This project relies heavily on secondary data. After that data splitting, data processing, and model development of NEAT and different machine learning algorithms and performance metrics evaluation have been mentioned in this section.

## 4.2 Secondary data

Secondary data is information previously collected by someone else for a different reason but is now being used by academics for a new study. It can be a helpful resource for academics because it reduces the time and cost of primary data collection. Published research such as books, academic journals, conference papers, online databases and repositories, websites and online platforms are some examples of secondary data.

## 4.3 Data Received for Research Work

Weekly pharmaceutical sales data is sourced from the Kaggle website [34]. This database comprises 600,000 sales records for 57 medicines from 2014-2019, including date, time, brand name, and quantity. This work predicts demand for specific classes of medications, rather than specific drugs. The selected 57 medications are divided into seven groups based on anatomical treatment chemical (ATC) theme analysis, as illustrated in Figure 6. Figure 7 displays sales patterns across all pharmacological dataset categories. It explains the dataset used for the investigation. Here in Table 1, we can see a sample of our data set.

**Table 1: Sample data for all drug categories**

| Date Tim | M01AI | M01A | N02BA | N02BE | N05B | R03 | R06 |
|---|---|---|---|---|---|---|---|
| 1/5/2014 | 14 | 12 | 21 | 186 | 41 | 32 | 7 |
| 1/12/2014 | 29 | 13 | 38 | 191 | 88 | 21 | 7 |
| 1/19/2014 | 31 | 26 | 46 | 218 | 80 | 29 | 12 |
| 1/26/2014 | 34 | 32 | 32 | 180 | 80 | 23 | 10 |
| 2/2/2014 | 31 | 23 | 21 | 160 | 84 | 29 | 12 |
| 2/9/2014 | 23 | 30 | 53 | 227 | 98 | 31 | 6 |
| 2/16/2014 | 33 | 31 | 47 | 321 | 89 | 18 | 16 |
| 2/23/2014 | 39 | 34 | 44 | 233 | 99 | 35 | 7 |
| 3/2/2014 | 37 | 29 | 40 | 223 | 52 | 16 | 10 |
| 3/9/2014 | 25 | 21 | 32 | 202 | 48 | 16 | 9 |
| 3/16/2014 | 32 | 24 | 36 | 165 | 90 | 36 | 19 |

### 4.3.1 Data Cleaning and Transformation

Data cleaning and transformation are essential steps to ensure the quality and relevance of data used in generating neural networks and in machine learning models. By thoroughly cleaning the data, handling missing values, removing duplicates, and data validation, we removed inaccuracies and inconsistencies that could mislead the model. Through transformation, we enhanced the dataset's structure and relevance, creating features that better represent the underlying patterns and improving the model's performance. These processes are vital for generating accurate and reliable demand forecasts in the pharmaceutical industry.

### 4.3.2 Adoption of the ATC Classification System

We used the World Health Organization's Anatomical Therapeutic Chemical (ATC) Classification System to categorize pharmaceutical goods sold in pharmacies over six years. This strategic classification was required not only to organize the data but also to ensure the interpretability and validity of our findings. We divided 57 different pharmaceutical goods into eight unique categories using the ATC medication classification system. This systematic categorization inside the ATC framework not only streamlined our data but also improved the clarity and correctness of our conclusions, considerably increasing the depth and usefulness of our pharmaceutical forecasting research.

| | |
|---|---|
| **M01AB** | Anti-inflammatory and antirheumatic products |
| **M01AE** | Non-steroidal anti-inflammatory drugs, propionic acid derivatives |
| **N02BA** | Analgesic, antipyretic, and salicylic acid derivative |
| **N02BE** | Anilides, analgesics, Paracetamol, Phenacetin |
| **N05B** | Anxiolytics, psycholeptics, Antihistamine drugs |
| **R03** | Drugs for obstructive airway diseases |
| **R06** | Antihistamines for systemic use |

**Figure 5: Components of Drug dataset**

## 4.4 Time Series Analysis

After preparing the data, we visually analyzed sales patterns using several visualizations. Time series charts have evolved into timelines that show sales trends over a certain era. The timelines described the cyclicality and variety in sales data by representing their temporal complexities. Figure 7 depicts the pattern of sales in all the drug categories. Prescription rates peak in the fourth quarter and fall in the first, which could be attributed to seasonal illnesses. Drug category R06 has the clearest troughs and peaks in the data pattern.

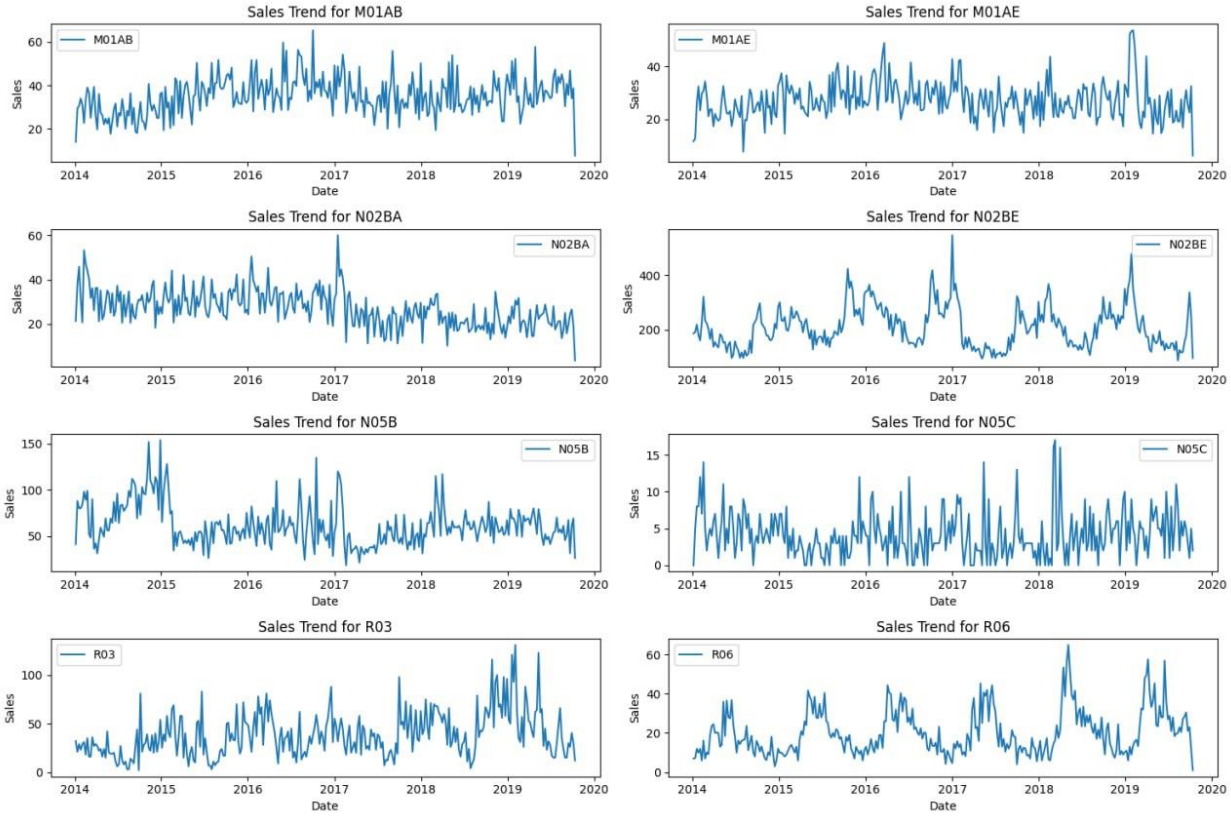


**Figure 6: Sales pattern of M01AB drug from 2014-2019**

### 4.4.1 Descriptive Analysis of Weekly Sales Data

Descriptive analysis provides a foundational understanding of your data, helping to summarize its main characteristics quantitatively. This includes calculating the mean, which provides the average value of the data, and the standard deviation (SD), which measures the spread or dispersion around the mean. The count indicates the number of non-null entries in the dataset. The minimum (min) and maximum (max) values identify the range within which the data points lie. These statistics help understand the dataset's central values, spread, and overall distribution, before deeper analysis or modeling.

**Table 2: Descriptive Statistics of the data**

| Statistic | M01AB | M01AE | N02BA | N02BE | N05B | R03 | R06 |
|---|---|---|---|---|---|---|---|
| **count** | 302.0 | 302.0 | 302.0 | 302.0 | 302.0 | 302.0 | 302.0 |
| **mean** | 35.1 | 27.17 | 27.06 | 208.63 | 61.74 | 38.44 | 20.22 |
| **std** | 8.62 | 7.04 | 8.09 | 76.07 | 22.44 | 22.9 | 11.38 |
| **min** | 7.67 | 6.24 | 3.5 | 86.25 | 18.0 | 2.0 | 1.0 |
| **max** | 65.33 | 53.57 | 60.12 | 546.9 | 154.0 | 131.0 | 65.0 |

## 4.5 Hyperparameter Search Space

The hyperparameter search space is the set of possible hyperparameter values searched throughout the hyperparameter tuning process to improve the performance of a machine learning model. Hyperparameters are settings or configurations that are defined before the learning process begins and are not derived from the data. The goal is to determine the hyperparameter combinations that produce the best-performing model.

**Table 3: Hyperparameters of the Models**

| | | |
|---|---|---|
| **Random Forest** | n-Estimator<br>Max_depth<br>Min_samples_split<br>Min_samples_leaf<br>Bootstrap | [50, 100, 150, 200, 300]<br>[None, 10, 20, 30]<br>[2, 5, 10]<br>[1, 2, 4]<br>[True, False] |
| **LSTM** | Number of features<br>Model type<br>Layers<br>Activation<br>Optimizer<br>Loss<br>Split Ratio | 1<br>Sequential<br>LSTM input layer (5,1)<br>'Relu'<br>Adam (Learning rate 0,0001)<br>'mse', 'mape'<br>.3 (30% validation, 70% train) |
| **XGBoost** | learning_rate<br>n_estimators max_depth<br>Subsample<br>colsample_bytree<br>colsample_bylevel<br>Gamma<br>Split Ratio | [0.001, 0.01, 0.05, 0.1, 0.5]<br>[30, 50, 100, 150, 200, 300]<br>[2, 3, 5, 7, 9, 11]<br>[0.7, 0.8, 0.9, 1]<br>[0.7, 0.8, 0.9, 1]<br>[0, 0.1, 0.2, 0.3, 0.4, 0.5]<br>.2 |
| **NEAT** | population_size<br>num_generations<br>fitness_threshold<br>activation<br>num_inputs<br>num_outputs | 150<br>100<br>0.9<br>'relu'<br>9<br>1 |

## Model Development

In model development for temporal data, creating lagged variables is a critical step in feature engineering that involves using previous time steps to predict future values. Lagged variables are generated by shifting the original time series data by a specific number of periods (lags). These lagged variables are then included as features in the dataset, allowing the model to learn temporal dependencies and improve forecasting accuracy. When working with weekly data and developing models using XGBoost, Random Forest, NEAT and LSTM, creating lagged variables becomes an essential feature engineering step.

### 4.5.1 Partial Autocorrelation Function

The Partial Autocorrelation Function (PACF) is a powerful tool in time series analysis for identifying the appropriate lag values to include in autoregressive models. By analyzing the PACF plot and identifying significant correlations, analysts can effectively determine the optimal lag structure for modeling temporal dependencies in their data, thereby improving the accuracy of time series forecasts.

Identifying significant spikes or peaks that exceed a certain confidence interval from Table 4, we find the lag values for all the drug categories.

**Table 4: Lag Values Determined from the PACF Graphs**

| Drug Category | M01AB | M01AE | N02BA | N02BE | N05B | R03 | R06 |
|---|---|---|---|---|---|---|---|
| Lag Value | 2 | 1 | 3 | 1 | 2 | 2 | 1 |

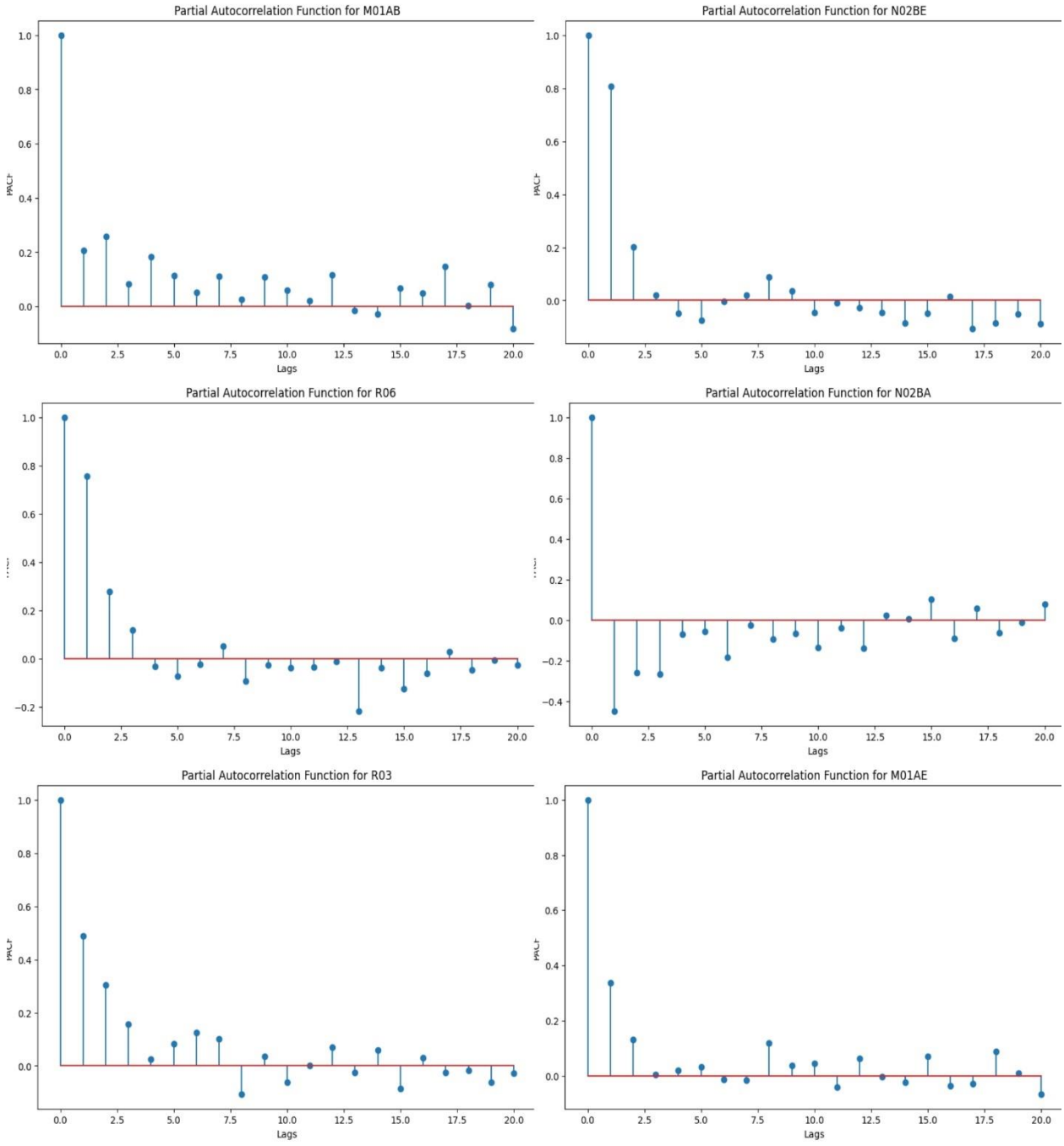


**Figure 7: Partial Autocorrelation Function Graphs for all the Drug Categories**

# Chapter 5 : Findings and Analysis

## 5.1 Introduction

In this chapter, findings from the developed neural network forecasting models NEAT and LSTM and traditional machine learning models XGBoost and Random Forest on our pharmaceutical temporal data, comparison among these models are described.

## 5.2 Actual Vs Predicted Graphs

The "Actual vs Predicted" graphs are essential visual tools used to assess the performance of predictive models, particularly in the context of time series forecasting. These graphs compare the actual observed values of the target variable with the predicted values generated by the model. Actual vs Predicted graphs are crucial in evaluating the performance and reliability of predictive models. They provide a clear visual representation of how well the model predicts future values compared to what occurs.

### 5.2.1 Actual Vs Predicted Graphs for XGBoost Model

Figure 9 demonstrates that the predicted values closely match the actual values for the drug categories R06 and N02BE, indicating a strong fit for the model. Similarly, for the drug category R03, the predicted values are also very close to the actual values, suggesting good model performance in this case as well.

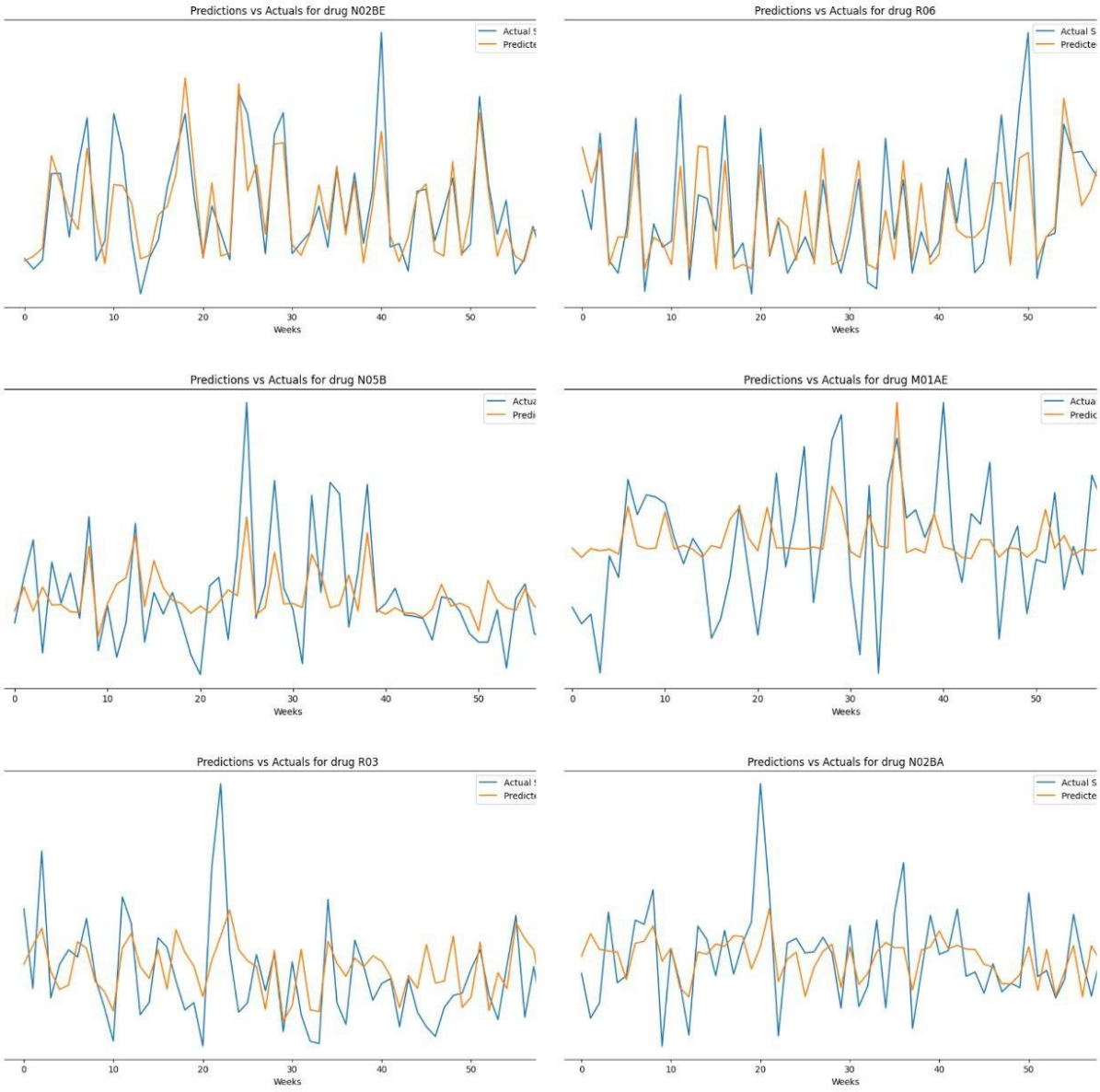


**Figure 8: Actual Vs Predicted Graph for XGBoost model**

### 5.2.2 Actual Vs Predicted Graphs for Random Forest Model

Developing the Random Forest model for all the drug categories, it is clearly visible that there is a strong fit of the model for drug categories N02BA, N02BE, and N05B. But for the drug category M01AB this model may not be ideal as predicted values deviate from the actual values

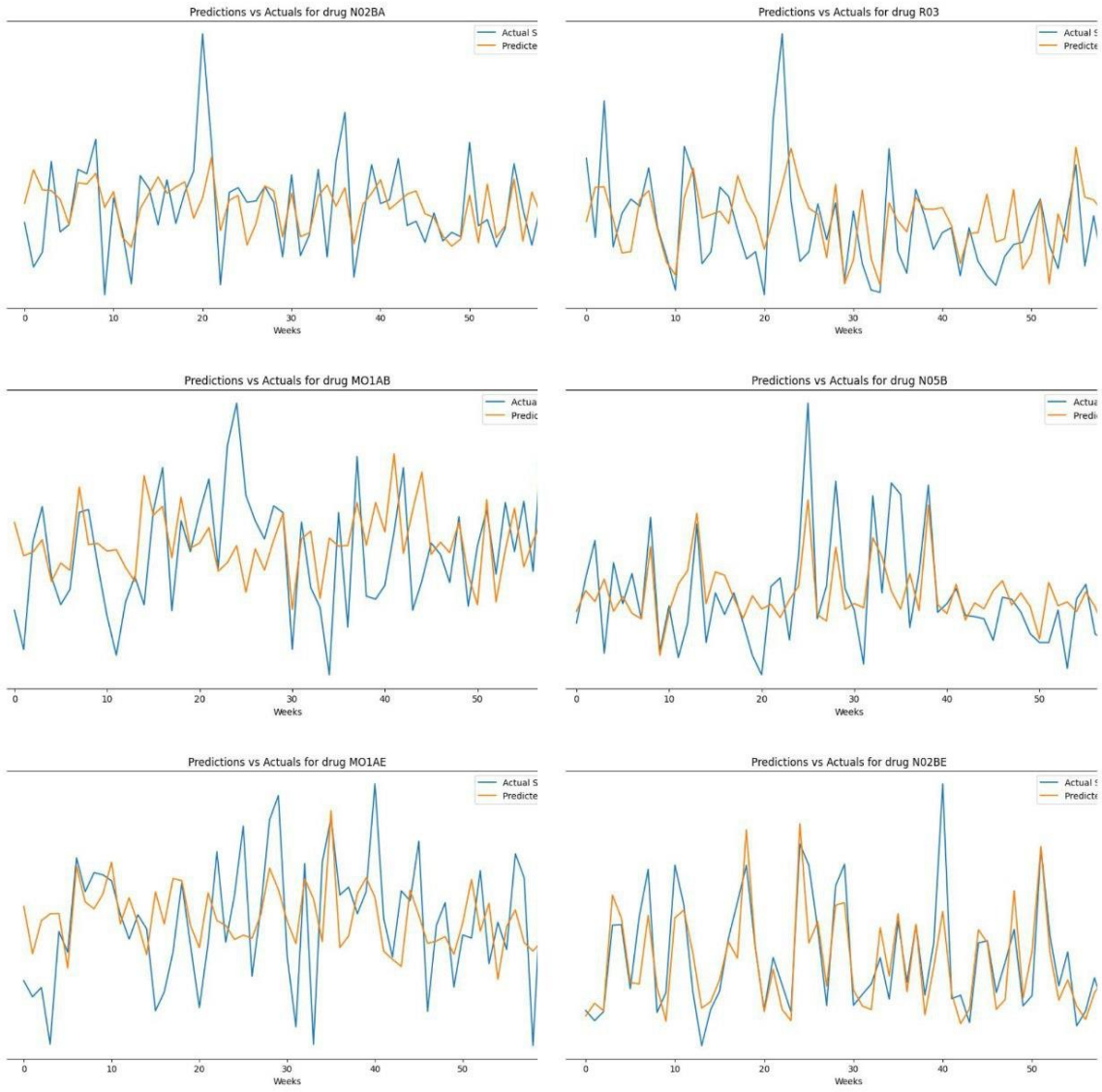


**Figure 9: Actual vs. predicted Graph for Random Forest model**

### 5.2.3 Actual Vs Predicted Graphs for Random LSTM Model

The prediction curve is a strong fit for drug categories R06 and N05B in the Long Short-Term Memory model but for N05 and M01AE categories, the prediction curve deviates from the actual curve.

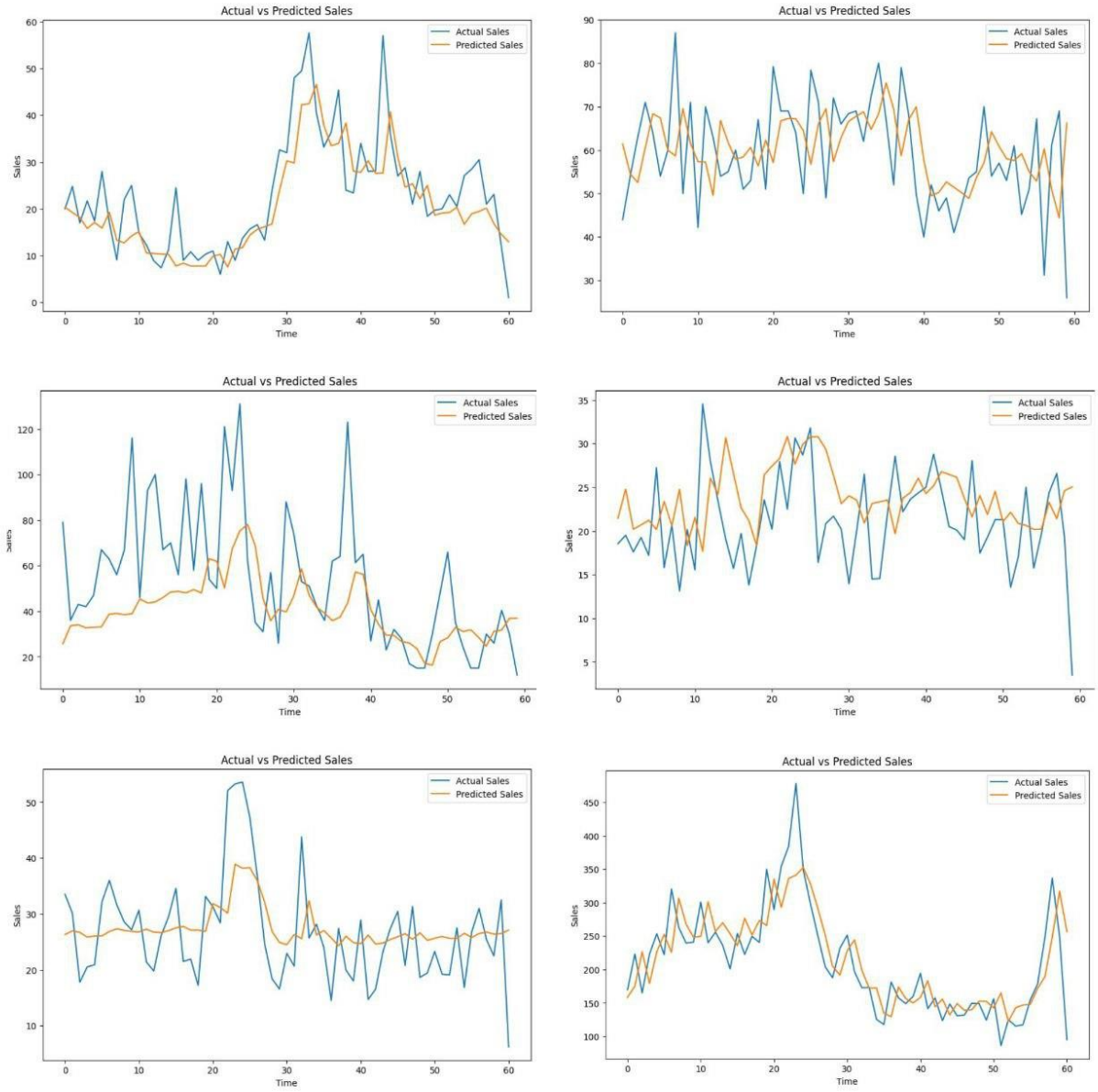


**Figure 10: Actual vs. predicted Graph for LSTM model**

### 5.2.4 Loss Curves for the LSTM model

A loss curve for an LSTM model is a graphical representation of how the loss (or error) changes over the course of training. This curve helps in understanding the training dynamics of the model and evaluating its performance.

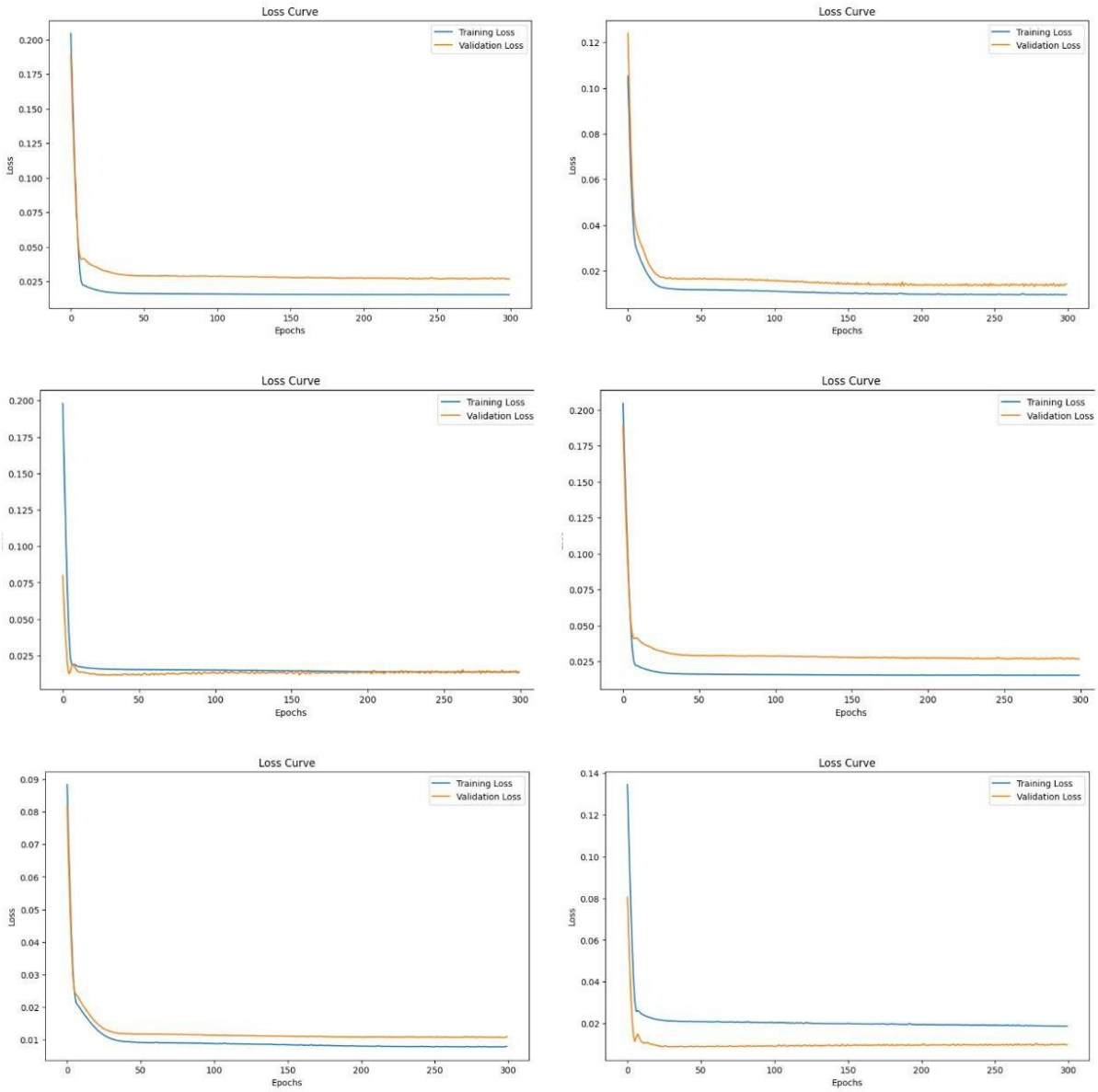


**Figure 11: Loss Curves for all the Drug Categories.**

### 5.2.5 Actual Vs Predicted Graphs for the NEAT Model

The NEAT model shows a strong fit curve for the R06 drug category and the predicted values are very close to the actual values for R06 evident from the curves but for other drug categories the curve deviates from the actual values.

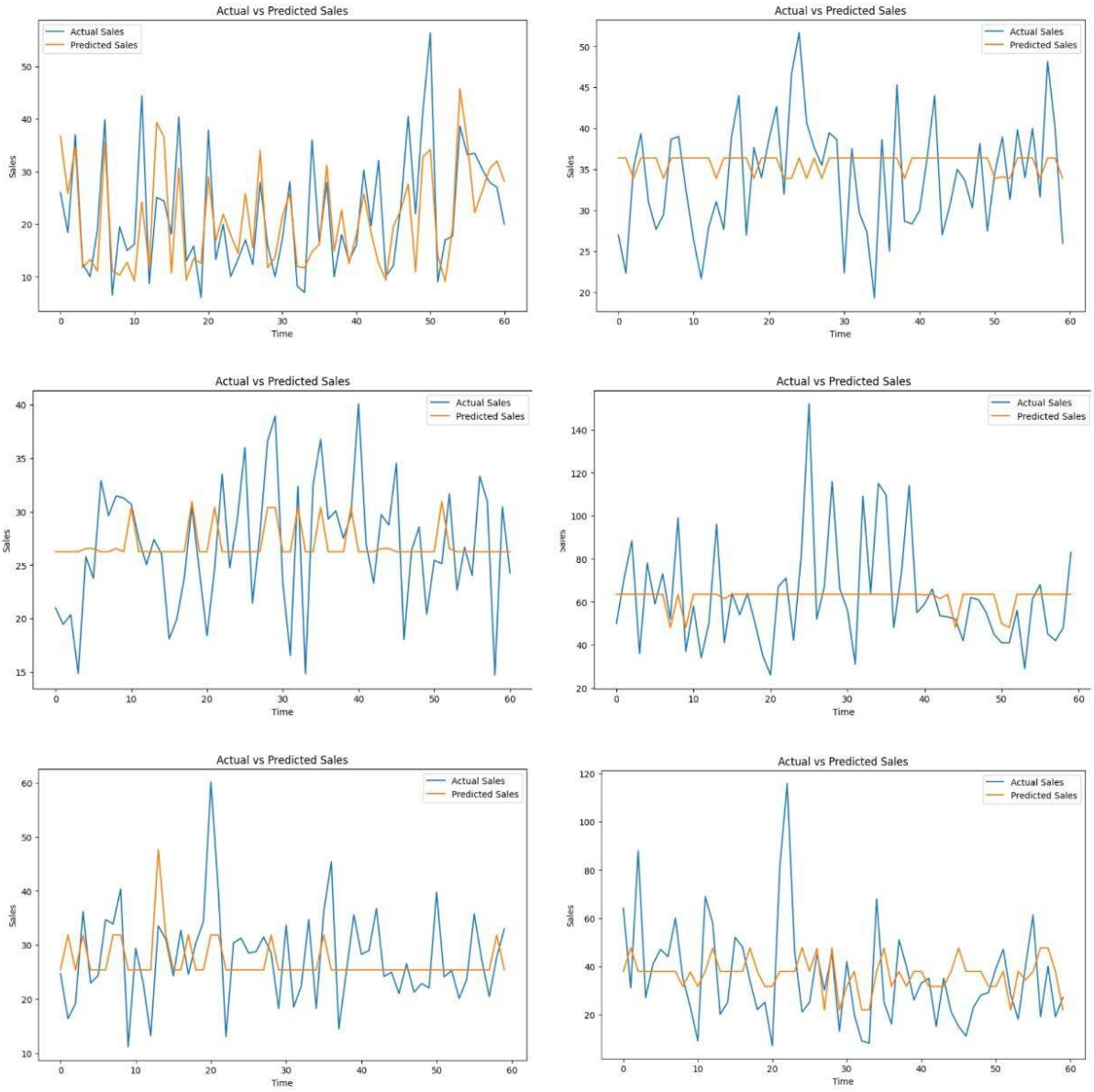


**Figure 12: Actual vs. Predicted Graphs for NEAT**

## 5.3 Evaluation of Performance Metrics

Evaluation of performance metrics involves assessing how well a machine learning model predicts outcomes by comparing its predictions to actual values. The choice of metrics depends on the type of problem. When evaluating forecasting models, RMSE (Root Mean Squared Error) and MAPE (Mean Absolute Percentage Error) are critical performance metrics. A lower RMSE indicates better model performance. MAPE, on the other hand, expresses the error as a percentage, providing an intuitive understanding of prediction accuracy relative to the actual values.

### 5.3.1 RMSE Values of the Models:

RMSE measures the average magnitude of the errors between predicted and actual values, with errors being squared before averaging, which emphasizes larger errors. A lower RMSE value indicates a better fit of the model to the data, signifying more accurate predictions.

**Table 5: Value of RMSE for all the Drug Categories**

| | M01AB | M01AE | N02BA | N02BE | N05B | R03 | R06 |
|---|---|---|---|---|---|---|---|
| **Random Forest** | 7.1675 | 5.6204 | 7.5992 | 39.021 | 19.4838 | 19.246 | 7.6772 |
| **XGBoost** | 7.045 | 5.439 | 7.414 | 35.713 | 19.927 | 18.245 | 7.028 |
| **NEAT** | 6.924 | 5.59 | 8.16 | 69.86 | 24.089 | 18.416 | 8.101 |
| **LSTM** | 7.997 | 7.739 | 6.645 | 48.23 | 13.446 | 28.717 | 7.657 |

From Table 5 it is evident that the XGBoost model has the lowest RMSE for most of the drug categories but LSTM and NEAT have also performed well for some drugs. For drug category M01AB, the NEAT model showed an RMSE value of only 6,924 outperforming the other categories. But for the N02BA drug category LSTM model showed the lowest value of RMSE. Now we can get a clear view at a glance of how all the models performed in a chart showing the RMSE value in a bar chart in Figure 14.

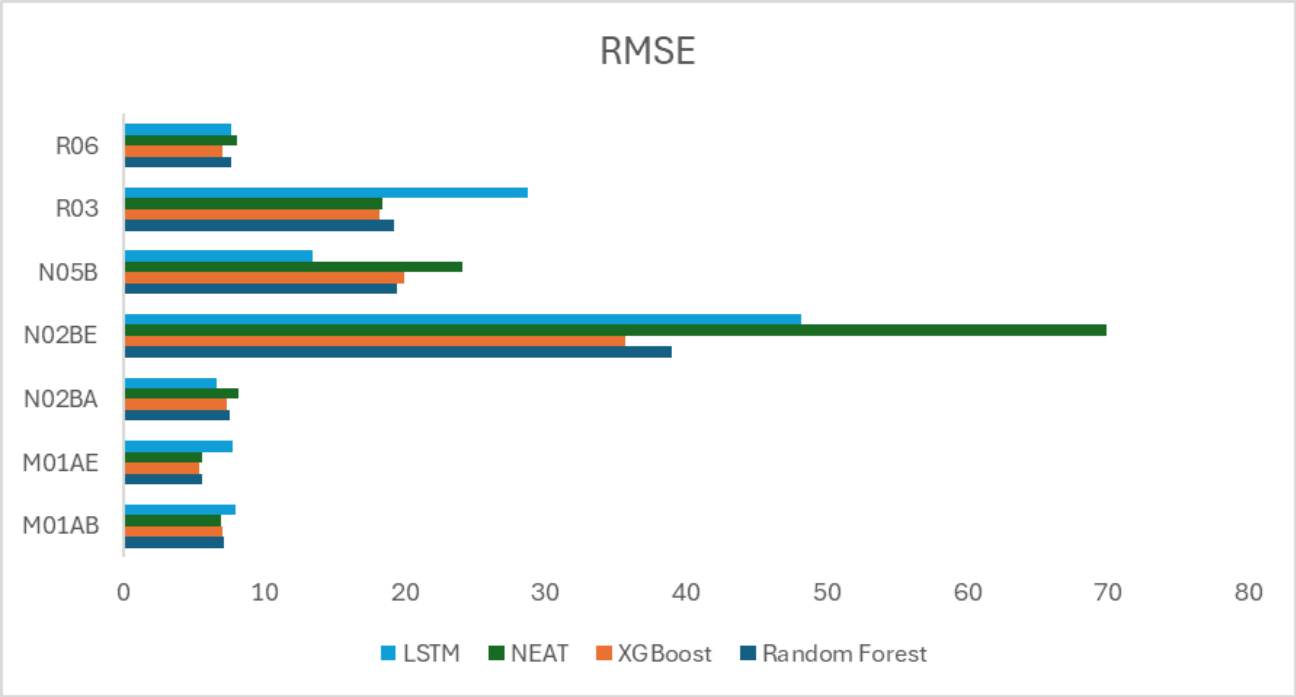


**Figure 13: Bar chart showing RMSE value for all the Drug Categories across the Models**

### 5.3.2 MAPE Values of the Models

MAPE measures the average absolute percentage difference between predicted and actual values, providing a normalized view of prediction errors. A lower MAPE value indicates a model that makes predictions closer to actual values in relative terms, which is particularly useful for comparing model performance across different datasets and scales.

**Table 6: Value of MAPE for all the Drug Categories**

| | M01AB | M01AE | N02BA | N02BE | N05B | R03 | R06 |
|---|---|---|---|---|---|---|---|
| **Random Forest** | 17.999 | 19.2966 | 22.8137 | 13.968 | 27.347 | 58.463 | 30.1518 |
| **XGBoost** | 17.73 | 18.655 | 22.432 | 13.4815 | 26.717 | 59.4026 | 27.926 |
| **NEAT** | 18.837 | 18.97 | 24.82 | 32.404 | 30.133 | 59.596 | 31.673 |
| **LSTM** | 20.285 | 29.413 | 35.207 | 19.303 | 20.397 | 38.564 | 41.617 |

From Figure 15, we see that for the N05B and R03 drug categories, LSTM shows the lowest MAPE value becoming the ideal choice of prediction for R03 and N05B groups. But for M01AB, M01AE, N02BA, N02BE and R06 categories, XGBoost model shows the lowest MAPE.

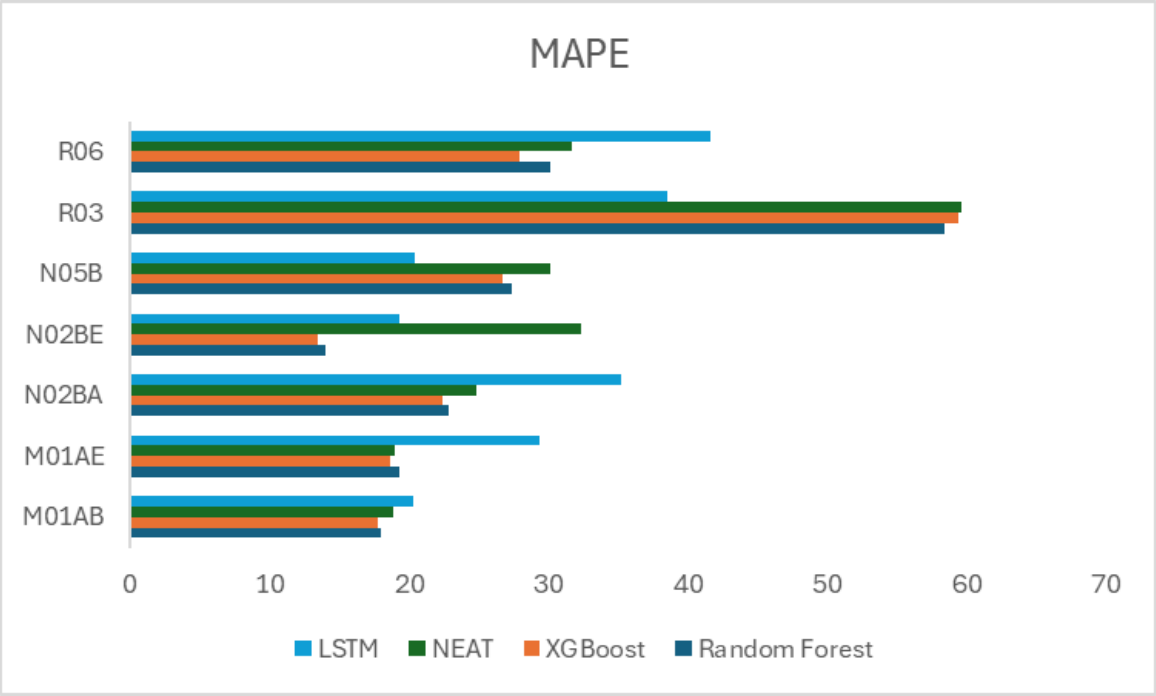


**Figure 14: Bar chart showing MAPE value for all the Drug Categories across the Models**

## 5.4 Discussion of Findings

In our analysis, the XGBoost model achieved the lowest RMSE and MAPE for drug categories R06, N02BE, and M01AE.The Random Forest model, while not the best overall, was the second-best option for categories such as R06, N02BE, and M01AB, with a MAPE of 13.96% for N02BE and an RMSE of 6.677 for R06. The LSTM model excelled for N05B, R03, and N02BA categories, making it ideal for predicting demand for psycholeptics, antihistamines, and antipyretic drugs, showing a MAPE of 20.397% for N05B and an RMSE of 6.65 for N02BA. The NEAT model showed the lowest RMSE for M01AB and M01AE categories and the second lowest MAPE for M01AE and R06, making it the best choice for non-steroidal anti-inflammatory drugs.

# Chapter 6 : Conclusion & Future Scopes

## 6.1 Conclusions

Our comprehensive analysis of various demand forecasting models for pharmaceutical sales has provided valuable insights and outcomes:

- The XGBoost model demonstrated superior performance with the lowest Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) for drug categories R06, R03, N02BE, and M01AE, encompassing treatments for respiratory diseases, pain relief, and systemic uses. However, it exhibited suboptimal results for respiratory disease drugs within the R03 category, suggesting variability in its effectiveness across this drug category.
- The Random Forest model, while not the top performer in any category, showed competitive performance as the second-best option for categories like R06, N02BE, and M01AB. This model's modest MAPE and RMSE values indicate its potential in specific drug categories despite not being the optimal choice overall.
- Transitioning to neural network-based models, LSTM (Long Short-Term Memory) emerged as highly promising for drug categories N05B, R03, and N02BA, achieving the lowest RMSE. This underscores LSTM's suitability for forecasting demand in psycholeptics, antihistamines, and antipyretics, highlighting its effectiveness with notably low MAPE values.
- NEAT (NeuroEvolution of Augmenting Topologies) model demonstrated compelling results by achieving the lowest RMSE for categories M01AB and M01AE, and the second lowest MAPE for M01AE and R06 drugs. Its application in forecasting non-steroidal anti-inflammatory drugs showcases NEAT as a strong contender where traditional machine learning models may falter.
- Our experimental findings confirm that no single forecasting model universally outperforms across all drug categories. Each model's efficacy varies depending on the

therapeutic area and type of pharmaceutical, necessitating tailored model selection based on specific forecasting needs. This nuanced approach ensures accurate predictions aligned with the diverse demands of pharmaceutical sales forecasting.

## 6.2 Future Scopes

Based on the findings and insights from our work, we would recommend exploring the following approaches to enhance the forecasting accuracy of pharmaceutical sales:

**1. Hybrid Models Integration:** There is scope of Integrating hybrid models such as Random Forest + LSTM and ARIMA + Neural Network to leverage the strengths of each component. Random Forest can be employed for robust feature selection, especially useful when incorporating diverse datasets like demographic or climate data. LSTM, known for capturing temporal dependencies in time series data, complements this by handling the sequential nature and long-term dependencies of pharmaceutical sales trends.

**2. Incorporation of External Datasets:** Extending the predictive capabilities by integrating external datasets such as demographic data (e.g., population trends, age distribution), climate data (e.g., weather patterns affecting health conditions), and promotional activity data (e.g., marketing campaigns, sales incentives) can provide additional contextual information. These datasets can enrich the feature set used by the models, thereby improving the precision of sales forecasts by accounting for broader environmental and market influences.

**3. Quantitative and Qualitative Feature Expansion**: Enhancing the model's predictive power by incorporating both quantitative and qualitative features can add more reliability and increase the efficiency in drug prediction. Quantitative features such as promotional activities and demographic trends provide measurable inputs for the models. Simultaneously, qualitative features like physician-patient reviews or broader health trends offer nuanced insights into consumer behavior and market dynamics. This dual approach ensures a comprehensive understanding of factors influencing pharmaceutical sales, allowing for more accurate predictions.

**4. Experimental Validation and Evaluation:** Conducting rigorous experimental studies will validate the effectiveness of these enhanced models. Evaluate performance metrics such as RMSE, MAPE, and accuracy across different datasets and time periods. This validation process will provide empirical evidence of the model's efficacy in handling the complexity and variability inherent in pharmaceutical sales forecasting.

**5. Continuous Improvement and Adaptation:** Emphasizing on continuous improvement by iterating on the models based on real-world feedback and updated data is the burning need for this sector. Pharmaceutical markets are dynamic, influenced by regulatory changes, market trends, and consumer behavior shifts. Regularly updating the models with new data and refining algorithms ensures their relevance and reliability in forecasting future sales scenarios.

In conclusion, by integrating hybrid models, leveraging external datasets, and expanding the feature set to include both quantitative and qualitative factors, our thesis proposes a robust framework for advancing pharmaceutical sales forecasting. These recommendations not only enhance prediction precision but also lay the groundwork for adapting to evolving market dynamics and improving strategic decision-making in the pharmaceutical industry.

# References


[1] Kumari A, Prasad U, Bala PK. Retail forecasting using neural network and data mining technique: a review and reflection. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS). 2013 Dec;2(6):266-9.

[2] Thomassey S. Sales forecasts in clothing industry: The key success factor of the supply chain management. International Journal of Production Economics. 2010 Dec 1;128(2):470-83.

[3] Kerkkänen A, Korpela J, Huiskonen J. Demand forecasting errors in industrial context: Measurement and impacts. International Journal of Production Economics. 2009 Mar 1;118(1):43-8.

[4] Wang L, Zeng Y, Chen T. Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Systems with Applications. 2015 Feb 1;42(2):855-63.

[5] Green KC, Armstrong JS. Simple versus complex forecasting: The evidence. Journal of Business Research. 2015 Aug 1;68(8):1678-85.

[6] Chu CW, Zhang GP. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. International Journal of Production Economics. 2003 Dec 11;86(3):217-31.

[7] Kuo RJ, Xue KC. A decision support system for sales forecasting through fuzzy neural networks with asymmetric fuzzy weights. Decision Support Systems. 1998 Dec 1;24(2):105-26.

[8] Chang PC, Lai CY, Lai KR. A hybrid system by evolving case-based reasoning with genetic algorithm in wholesaler's returning book forecasting. Decision Support Systems. 2006 Dec 1;42(3):1715-29.

[9] Ahn H, Choi E, Han I. Extracting underlying meaningful features and canceling noise using independent component analysis for direct marketing. Expert Systems with Applications. 2007 Jul 1;33(1):181-91.

[10] Tang Z, De Almeida C, Fishwick PA. Time series forecasting using neural networks vs. Box-Jenkins methodology. Simulation. 1991 Nov;57(5):303-10.

[11] Stock, J. and Watson, m. (1999), ―A Comparison of Linear and Nonlinear University Models for Forecasting Economic Time Series‖, In R. F. Engle & H. White (Eds.), Cointegration, causality, Oxford: Oxford University Press. pp. 1–44.

[12] Ravi, G.M. (2005), ―Forecasting Technique, Industrial Statistics and Operational Management, pp.142-143.

[13] Kress, G. (1985) Forecasting Courses for Managers. C.L. Jain, ed. Graceway Publishing Company, New York.

[14] Kahn, K. B., & Mentzer, J. T. (1994). The impact of team-based forecasting. *The journal of business forecasting*, *13*(2), 18.

[15] Jobber, D. and Lancaster, G. (2009). Selling and Sales Management. 8th ed. Harlow: Prentice Hall, pp- 483.

[16] Thomassey, S. (2010). Sales forecasts in clothing industry: The key success factor of the supply chain management. *International Journal of Production Economics*, *128*(2), 470-483.

[17] Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods. *Journal of retailing and consumer services*, *8*(3), 147-156.

[18] Ma, S., Fildes, R., & Huang, T. (2016). Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra-and inter-category promotional information. *European Journal of Operational Research*, *249*(1), 245-257.

[19] Schneider, M. J., & Gupta, S. (2016). Forecasting sales of new and existing products using consumer reviews: A random projections approach. *International Journal of Forecasting*, *32*(2), 243-256.

[20] Soldo, B.I., (2012), ―Forecasting Natural Gas Consumption‖, Appl Energy, Vol. 92, pp.26–37.

[21] Azadeh, A., Saberi, M., Asadzadeh, S. M., Hussain, O. K., & Saberi, Z. (2013). A neuro-fuzzy-multivariate algorithm for accurate gas consumption estimation in South America with noisy inputs. *International Journal of Electrical Power & Energy Systems*, *46*, 315-325.

[22] van der Vorst, J. G., Beulens, A. J., de Wit, W. V., & van Beek, P. (1998). Supply chain management in food chains: Improving performance by reducing uncertainty. *International Transactions in Operational Research*, *5*(6), 487-499

[23] Sha, W., & Edwards, K. L. (2007). The use of artificial neural networks in materials science based research. *Materials & design*, *28*(6), 1747-1752.

[24] Mjalli, F. S., Al-Asheh, S., & Alfadala, H. E. (2007). Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance. *Journal of Environmental Management*, *83*(3), 329-338.

[25] Abraham, A. (2004). Meta learning evolutionary artificial neural networks. *Neurocomputing*, *56*, 1-38.

[26] Zhang, G. P. (Ed.). (2004). *Neural networks in business forecasting*. IGI global.

[27] Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

[28] Mandic, D. P., & Chambers, J. (2001). *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. John Wiley & Sons, Inc.

[29] https://neat-python.readthedocs.io/en/latest/neat_overview.html

[30] Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, *10*(2), 99-127.

[31] Mahfoud, S. W. (1995). *Niching methods for genetic algorithms*. University of Illinois at Urbana-Champaign.

[32] Ibrahim, M. Y., Sridhar, R., Geetha, T. V., & Deepika, S. S. (2019, December). Advances in neuroevolution through augmenting topologies–a case study. In *2019 11th international conference on advanced computing (ICoAC)* (pp. 111-116). IEEE.

[33] Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). Traffic flow prediction with big data: A deep learning approach. *Ieee transactions on intelligent transportation systems*, *16*(2), 865-873.

[34] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, *18*(5-6), 602-610.

[35] Carbonneau, R., Vahidov, R., & Laframboise, K. (2007). Machine learning-based demand forecasting in supply chains. *International journal of intelligent information technologies (IJIIT)*, *3*(4), 40-57.

[36] Zhang, L., Bian, W., Qu, W., Tuo, L., & Wang, Y. (2021, April). Time series forecast of sales volume based on XGBoost. In *Journal of Physics: Conference Series* (Vol. 1873, No. 1, p. 012067). IOP Publishing.

[37] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[38] Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929-5955.

[39] Ren, Q., Cheng, H., & Han, H. (2017, March). Research on machine learning framework based on random forest algorithm. In *AIP conference proceedings* (Vol. 1820, No. 1). AIP Publishing.

[40] Provost, F., Hibert, C., Malet, J. P., Stumpf, A., & Doubre, C. (2016, April). Automatic classification of endogenous seismic sources within a landslide body using random forest algorithm. In *EGU general assembly conference abstracts* (pp. EPSC2016-15705).

[41] Bradter, U., Kunin, W. E., Altringham, J. D., Thom, T. J., & Benton, T. G. (2013). Identifying appropriate spatial scales of predictors in species distribution models with the random forest algorithm. *Methods in Ecology and Evolution*, *4*(2), 167-174.

[42] Ibrahim, M. Y., Sridhar, R., Geetha, T. V., & Deepika, S. S. (2019, December). Advances in neuroevolution through augmenting topologies–a case study. In *2019 11th International Conference on Advanced Computing (ICoAC)* (pp. 111-116). IEEE.

[43] MilanzdravKovic (2020) Pharma sales data analysis and forecasting. Available: https://www.kaggle.com/milanzdravkovic/pharma-sales-data-analysis-and-forecasting/data

[44] Feng, Y., & Wang, S. (2017, May). A forecast for bicycle rental demand based on random forests and multiple linear regression. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)* (pp. 101-105). IEEE.

[45] Goldberg, D. E., & Richardson, J. (1987, July). Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms* (Vol. 4149, pp. 414-425).

**END**