TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

Thesis No.: PUL080MSCoM008

ANN based Prediction of Final Construction Cost of Residential Buildings in Kathmandu Valley at an Early Stage

By

Bikram Pathak

A THESIS

SUBMITTED TO THE DEPARTMENT OF CIVIL ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER IN CONSTRUCTION MANAGEMENT

DEPARTMENT OF CIVIL ENGINEERING

LALITPUR, NEPAL

April 2026

COPYRIGHT

The author has agreed that the library, Department of Civil Engineering, Pulchowk Campus, and Institute of Engineering may make this thesis freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this thesis for scholarly purposes may be granted by the professor, who supervised the work recorded herein or, in their absence, by the Head of the Department wherein the thesis was done. It is understood that recognition will be given to the author of this thesis and to the Department of Civil Engineering, Pulchowk Campus, Institute of Engineering for in any use of the material of this thesis. Copying or publication or the other use of this thesis for financial gain without approval of the Department of Civil Engineering, Pulchowk Campus, Institute of Engineering and author's written permission is prohibited.

Request for permission to copy or to make any other use of the material in this thesis in whole or in part should be addressed to:

Head of Department

Department of Civil Engineering

Pulchowk Campus, Institute of Engineering

Lalitpur, Nepal

DECLARATION

I hereby declare that the thesis entitled "**ANN based Prediction of Final Construction Cost of Residential Buildings in Kathmandu Valley at an Early Stage** " submitted to the Department of Civil Engineering in partial fulfillment of the requirement for the degree is Masters in Science in Construction Management, is a record of a work done under the supervision of Mr. Subash Kumar Bhattarai, Institute of Engineering, Pulchowk Campus. The thesis contains only my work except for the consulted material, which has been duly referenced and acknowledged.

....

Bikram Pathak

PUL080MSCoM008

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF CIVIL ENGINEERING

Tribhuvan University
Institute of Engineering
Department of Civil Engineering
Pulchowk Campus

CERTIFICATE OF THESIS APPROVAL

The undersigned certify that we have read and recommended to the Institute of Engineering for acceptance, a thesis entitled "**ANN based Prediction of Final Construction Cost of Residential Buildings in Kathmandu Valley at an Early Stage**" submitted by Bikram Pathak in partial fulfillment of the requirements for the degree of Master of Science in Construction Management.

Supervisor:

....

Er. Subash Kumar Bhattarai
Visiting Faculty
Construction Management
Centre for Post Graduate Studies
Nepal Engineering College (nec-CPS)

External Examiner:

....

Dr. Dinesh Sukamani
M.Sc. in Construction Management
Program Coordinator,
Nepal Engineering College (NEC)
Prayagpokhari, Lalitpur

Program Coordinator:

....

Asst. Prof. Mahendra Raj Dhital
M.Sc. in Construction Management
Department of Civil Engineering
IOE, Pulchowk Campus

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to my thesis supervisor, Mr. Subash Kumar Bhattarai for his guidance, insightful feedback and constant encouragement throughout the study that enabled this study to take its final form. His expertise and constructive criticism have been invaluable in shaping this thesis. I am also immensely thankful to Asst. Professor Mahendra Raj Dhital, the Program Coordinator, for his continuous support and for providing me with the opportunity to pursue this thesis as part of the Master of Science in Construction Management program.

I extend my sincere appreciation to all the experts who have shared their invaluable insights and knowledge, which greatly enriched the quality of this thesis. Their contributions have been instrumental in the successful completion of this work. I would also like to acknowledge the assistance of my colleagues and the construction firms for their help in data collection and coordination.

Finally, I am extremely grateful to my family and friends for their continuous support, patience, and encouragement throughout this journey. Their belief in me has been a constant source of motivation. I am also deeply grateful to my fellow engineer, friend Mr. Bibek Gyawali, Mr. Aabiskar Marasini and my brother Mr. Bidhan Pathak for their valuable support and assistance during data collection and analysis. Without the collective efforts and contributions of all these individuals, this thesis would not have been possible. I sincerely thank everyone who has been part of this endeavor, directly or indirectly.

Bikram Pathak

080MSCoM008

`

ABSTRACT

In Nepal, the construction industry contributes approximately 10-11% of the national GDP yet continues to suffer from persistent cost overruns and schedule delays, largely attributed to the limitations of traditional estimation methods such as regression models and quantity rate analysis. This research proposes the development of an Artificial Neural Network (ANN)-based model for predicting the final construction cost of residential buildings in Kathmandu Valley at an early stage of project development. The model used early-stage project parameters, including structural and design-related factors, as inputs to train a feedforward neural network with backpropagation. Historical data from completed building projects in Kathmandu Valley was collected, preprocessed, and adjusted for cost inflation before being used to train and validate the model on an 85-15 data split. The results revealed that the developed robust ANN model (10-128-64-1) achieved a good predictive accuracy, with a correlation coefficient (R^2) of 0.9019 and 0.6787 on training and test set respectively. A training Mean Absolute Percentage Error (MAPE) of 4.97% and a test MAPE of 7.32%, which was further validated through 10-fold cross validation with average R^2 of 0.5211 and average MAPE of 7.50%, with the best model (model 5) having a MAPE value of 5.90% demonstrated high accuracy of the model proving its robust generalization capabilities, suggesting that it can serve as a dependable decision-support tool even when provided with limited conceptual data. Thus, the model provides a locally calibrated framework to help stakeholders in Nepal make informed financial decisions during the project feasibility phase.

Keywords *Artificial Neural Network, Cost Estimation, Construction Management, Kathmandu Valley, Machine Learning, Backpropagation*

`

TABLE OF CONTENTS

`

`

LIST OF FIGURES

LIST OF TABLES

`

ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
BOQ	Bill of Quantities
BPNN	Backpropagation Neural Network
CM	Construction Management
CSV	Comma Separated Values
DUDBC	Department of Urban Development and Building Construction
GDP	Gross Domestic Product
HVAC	Heating, Ventilation, and Air Conditioning
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLPNN	Multilayer Perceptron Neural Network
MOUD	Ministry of Urban Development
MSE	Mean Squared Error
NBC	Nepal National Building Code
NRB	Nepal Rastra Bank
RCC	Reinforced Cement Concrete
SSR	Sum of Squared Residuals
TSS	Total Sum of Squares
WPI	Wholesale Price Index

CHAPTER 1: INTRODUCTION

1.1 Background

The global construction industry, worth at present 14.4 trillion USD, accounts for 14% of the global GDP and is characterized by a significant number of projects sporting delays and cost overruns, along with conflicts amongst stakeholders (Audisio, 2023). The construction industry forms a major portion of Nepal's economic development as well, which has been contributing substantially to the Nepal's GDP (Gross Domestic Product). It is second to agriculture in the number of employment opportunities provided to the people. According to the research by Santosh Baral, the Nepalese Construction Industry contributes around 10 to 11 percentage to the GDP and it uses around 35 percent of the government budget (Baral, 2009). Accurate cost estimation at the early stages of a building project is a crucial factor for successful project planning, budgeting, and decision-making in a construction project. Managing the project's costs is the most significant aspect of executing construction projects. Costs are considered primary criteria in the projects' feasibility studies and decision-making at the early stages of the project (Al-Gahtani, 2024). Nevertheless, there are a lot of difficulties faced by this sector such as making reliable predictions about construction costs that are very important when it comes to proper planning, resource allocation and financial management. According to the digitization report released by McKinsey, the construction industry is currently one of the worst-performing industries in terms of digitalization (Agarwal, Chandrasekaran, & Sridhar, 2016). Inaccurate construction cost estimates can cause delays in projects, overrun of budget, and ultimately, reduce the quality of the project and its profitability. For decades, researchers have been working to reduce the variance between the cost estimate and to actual one. However, as the project takes years to complete and labor, equipment, and raw material prices are expected to fluctuate in the unstable market, cost estimation can have a significant impact on the project's success (Alrasheed, 2025). According to the survey conducted by Ibrahim and Elshwadfy (2021), it was found that the top factors that influence the quality of the cost estimation process are design parameters and information, external factors include the materials' prices, availability, and importing process, and contractor performance.

Early-stage cost estimate plays a significant role in any initial construction project decisions, even though the project scope has not yet been finalized and very limited information is available regarding the detailed design during early stages (Arafa & Alqedra, 2011). Traditional methods are being used for preliminary cost estimation in the construction industry of Nepal so there still exists the problem of cost overrun, and time delay due to incorrect cost budgeting (Sapkota, Karki, Pokharel, & Dhital, 2023). This study aims at developing an efficient model to estimate the cost of building construction projects at early stages using artificial neural networks. The adoption of Artificial Neural Networks (ANN) represents a shift from traditional linear modeling to a more sophisticated, "brain-inspired" computational approach. At its core, an ANN consists of interconnected artificial neurons organized into input, hidden, and output layers. The learning process occurs within the hidden layers, where each neuron calculates a weighted sum of its inputs plus a bias, then applies a non-linear activation function, such as sigmoid or ReLU, to produce an output (Graupe, 2013). This mathematical relationship is typically expressed as:

$$y = f\,(W.\,x + b)$$

This ability to model complex, non-linear dependencies is particularly important in Kathmandu Valley, encompassing the districts of Kathmandu, Bhaktapur, and Lalitpur. Residential construction in this region is dominated by Reinforced Cement Concrete (RCC) frame structures designed according to the Nepal National Building Code (NBC 205). However, the valley's rapid and often dense urbanization introduces significant variability in costs due to differing land gradients, varying building designs, and localized fluctuations in material transportation and labor mobilization. By training a feedforward network with backpropagation, this research seeks to capture these intricate local variables to provide a reliable predictive tool where traditional quantity rate analysis often falls short. The Figure 1 below shows a diagram of a biological neuron and a node in Artificial Neural Network

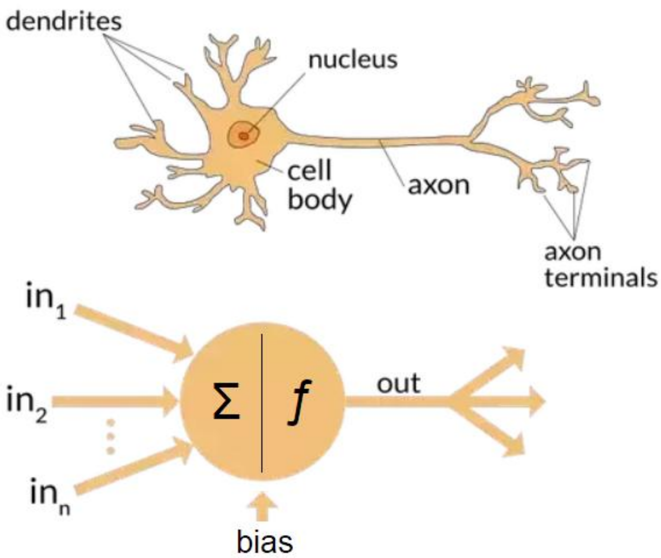

Figure 1 Biological Neurons and ANN

1.2 Problem Statement

The existing methodologies for predicting building construction costs in Nepal primarily rely on traditional regression models, which may not capture the complex, non-linear relationships inherent in construction projects. Cost estimation methods have evolved significantly over the past few decades, transitioning from experience-based judgment to statistical models and, more recently, to machine learning techniques. Understanding the limitations of traditional approaches provides the theoretical justification for adopting ANN in this study. Traditional parametric estimating relies on

`

correlating project cost to measurable physical characteristics such as floor area or number of rooms using regression equations. While regression models are straightforward to develop and interpret, they assume a linear relationship between input parameters and project cost. It is an assumption that is frequently violated in real-world construction data (Kim, An, & Kang, 2004). They demonstrated that while regression analysis requires explicit specification of the functional form relating variables to cost, neural networks can autonomously identify those relationships, giving them a significant advantage in capturing non-linear dependencies. Kim et al. (2004), comparing regression analysis, neural networks, and case-based reasoning using 530 historical cost records, found that the neural network model consistently outperformed the other two methods in prediction accuracy. More recent comparisons have extended this finding to a wider range of machine learning algorithms. A key limitation of ANN models is the so-called "black-box" nature of the prediction i.e.; the internal workings of the model are not transparent and the contribution of individual input variables to the prediction outcome is not immediately interpretable by practitioners. Despite this, the demonstrated accuracy improvements over traditional methods justify their adoption in practical cost estimation contexts.

For Nepal's construction industry, accurate cost estimation is an appealing option since it can result in possible cost savings and increased time efficiency during project execution. Like many other developing nations, Nepal frequently experiences delays and budget overruns as a result of erroneous early cost estimates. Growing urbanization, notably in the Kathmandu valley, has resulted in a rise in residential construction activity, putting further strain on accurate pre-construction cost estimation. Kathmandu valley includes three districts, namely, Kathmandu, Bhaktapur, and Lalitpur. These districts have been growing with huge number of residential buildings being built every year, which raises the question about the accuracy of cost estimated at the feasibility study stage of project. Reinforced Cement Concrete (RCC) building construction is the conventional building construction practice in Nepal, most specifically inside Kathmandu Valley (Chaulagain, Rodrigues, Spacone, & Varum, 2015). Despite the growing popularity of machine learning in various industries, its application to the construction sector in Nepal has been relatively limited (Sapkota, Karki, Pokharel, & Dhital, 2023). Quantity Estimation Methods are the primary conventional method for estimating costs that is commonly utilized for construction projects (Veliyampatt). Due to a number of significant variables and non-linear relationships between these variables, traditional approaches have limitations in accurate project cost prediction. To overcome the shortcomings of traditional cost estimation methods, this research proposes the development of a machine learning-based model specifically, an artificial neural network (ANN) for early-stage cost estimation of residential buildings in Kathmandu valley. Recent studies have shown that ANN outperforms multiple regression analysis and other traditional methods (Shamim, Hamid, Nyamasvisva, & Rafi, 2025).

1.3 Research Questions

The scope of this study is defined by the following research questions:

RQ1: What are the main factors influencing the construction cost of a building and how can these factors be validated in the case of Kathmandu valley?

`

RQ2: What is the relationship between the identified variables that affect construction costs of buildings in Kathmandu valley?

- RQ3: How to develop and validate an ANN model capable of predicting construction costs of building projects efficiently?

1.4 Research Objectives

The main objective of this research study is to develop an ANN-based prediction model capable of predicting construction costs of residential building projects in Kathmandu Valley using a machine learning technique called Artificial Neural Network which will be accomplished through the following specific objectives.

- To identify the main factors influencing the construction cost of a building.
- To find the relationship between identified variables that affect the construction costs of buildings in Kathmandu valley.
- To develop an ANN model capable of predicting construction costs of building projects efficiently.

1.5 Significance of the study

The construction sector, which accounts for 10–11% of the country's GDP, is essential to Nepal's economic growth. However, the sector is frequently hindered by cost overruns and schedule delays, largely due to the limitations of traditional estimation methods. This study is significant for several reasons:

- **Improved Accuracy in Early-Stage Planning**: Accurate cost estimation at the project conception phase is vital for successful budgeting and feasibility. This research provides an Artificial Neural Network (ANN) model with a high correlation coefficient, thus offering a more reliable alternative to traditional methods that often struggle with the non-linear complexities of construction data.

- **Localized Decision-Support Tool**: By focusing specifically on the Kathmandu Valley, the study addresses unique regional challenges such as urban density, district-specific land costs, and site accessibility. It serves as a practical decision-support tool for stakeholders, including investors, project managers, and consultants. Therefore, allowing them to make informed financial decisions even with limited conceptual data.

- **Advancing Digitalization in Nepal**: Despite the global rise of machine learning, its application in Nepal's construction sector has remained limited. This research bridges that gap by demonstrating how modern data-driven approaches can outperform experience-based judgment and manual quantity estimation.

`

- **Economic Efficiency**: By reducing the variance between estimated and actual costs, the model helps minimize budget overruns and project delays. This ensures better allocation of limited capital resources, ultimately enhancing the profitability and quality of residential building projects.

1.6 Research Scope and Limitations

In Nepal, where traditional cost estimation relies predominantly on quantity rate analysis and expert judgment, inaccurate forecasts continue to cause budget overruns, delays, and reduced project feasibility. This study addresses that challenge by proposing a feedforward Artificial Neural Network (ANN) model trained on historical project data from the Kathmandu Valley providing a focused, locally grounded alternative to the generalized estimation tools currently in use. The choice of ANN as the prediction methodology is firmly supported by a substantial body of international and regional literature. Data of 76 residential buildings completed since 2023 were collected from construction firms and consultants specializing in the field of building construction. The model will provide accurate duration estimates based on key parameters, thereby offering valuable insights during the early stages of project development. By using this model, stakeholders can make more informed decisions regarding project timelines, ultimately enhancing efficiency and accuracy in construction project management.

After thorough literature review and expert validation, the model incorporates a range of input variables: number of storeys, plinth area, floor height, and number of columns as numerical inputs; and location (district), foundation type as categorical inputs, to address the variables that affect cost estimation in residential buildings. By considering these diverse factors, the model aims to provide a more comprehensive and data-driven alternative to the traditional plinth area estimation methods currently used by contractors, and designers alike. The scope of this research is limited to residential buildings in Kathmandu Valley and is intended to support preliminary budgeting efforts rather than detailed engineering cost analyses conducted in later project stages. The scope of the study is intentionally limited to factors governing the structural cost of buildings. Cece Suhendi (2024), in their study state that structural components typically account for approximately one-third of total residential building construction cost and structural parameters available at the early design stage have been shown to be sufficient for generating reliable cost predictions in multiple published studies (Alrasheed, 2025).

The limitations of this study approach are as follows:

1) The model is trained on a dataset of only 76 residential building projects, which may limit its ability to generalize to broader or more diverse building types and conditions.
2) Since the model is trained exclusively on building projects from three districts of Kathmandu valley, its performance may not translate effectively to regions

outside the valley or to future projects that differ significantly from the historical data used in training.

3) Residential buildings constructed in Kathmandu valley are generally Reinforced Cement Concrete (RCC) structures, thus the scope of the study is limited to as such and not applicable to steel, load bearing structures.
4) The model's input variables focus on structural parameters as the primary cost drivers, implying that the cost related to non-structural components of buildings may not explicitly be captured in the final predicted cost.
5) This model is intended solely for use during the early stages of project planning and the costs predicted by the model are the expected costs to be incurred during construction phase so the costs pertaining to the design and pre-construction phase are not applicable to this research.

CHAPTER 2: LITERATURE REVIEW

2.1 Buildings definition

Any structure constructed of any building material, including the foundation, plinth, walls, floors, roof, and building services, is referred to as a building, regardless of whether it is inhabited by people. Public buildings are any governmental, non-governmental, and private structures that provide public opportunities, services, facilities, and goods. Buildings are categorized as residential, commercial, educational, hospital, office, industrial, storage, and assembly based on utilization. Residential buildings constitute the largest share of the construction sector in developing countries, both in terms of volume and investment (Karn & Dahal, 2021). In Nepal, rapid urbanization: particularly in the Kathmandu valley, has led to a surge in residential construction activity, placing increased pressure on accurate pre-construction cost estimation. The Kathmandu valley encompasses three districts, namely, Kathmandu, Bhaktapur, and Lalitpur. Residential buildings in Nepal are predominantly reinforced concrete (RCC) frame structures, designed in compliance with the Nepal National Building Code (NBC 205), and vary considerably in terms of the number of storeys, plinth area, foundation type, and structural system employed (Karn & Dahal, 2021).

2.2 Cost Estimation in Construction Projects

Cost estimation is a fundamental process in construction project management, involving the forecasting of time, cost, and resources necessary to achieve project objectives (Alrasheed, 2025). Dysert defines cost estimation as the systematic process of quantifying financial resources and preparing a project budget based on the defined scope of investment (Dysert, 2008) . Similarly, Hatamleh, et al., (2018) describe it as identifying the scope of work and the associated financial resources required from project initiation to handover. Many factors affect construction project costs, and some present significant challenges in the cost forecasting process. The reliability of the prediction approach determines whether the prediction models are successful or unsuccessful (Moghayedi, 2022). Various techniques and methods are available to develop predictive models of relationships among various variables in the construction industry, such as regression analysis methods, time series analysis, and machine learning techniques. Machine learning techniques such as neural networks and multilayer perceptron have demonstrated better performance in cost forecasting in comparison to statistical multiple regressions (Magdum, 2017).

Cost estimation methods in the construction industry are broadly categorized into two groups: traditional methods and modern data-driven approaches. Traditional methods include unit rate analysis, bill of quantities (BOQ), analogous estimating, and parametric estimating. Among these, quantity rate analysis, which involves calculating costs from detailed material quantities and unit rates, is the most commonly used method in Nepal (Sapkota, Karki, Pokharel, & Dhital, 2023).While these approaches can be precise when full design information is available, they are time-consuming and require detailed drawings and specifications that are not yet finalized at the early stages of a project. This creates a fundamental challenge: the most critical decisions about project viability and budgeting must be made precisely when the least information is available. The consequence of inaccurate early-stage cost estimation is most visible through cost overruns, which represent one of the most persistent and damaging problems in the construction industry worldwide. Studies spanning more than seven decades consistently show that construction projects exceed their initial budgets, with the average global cost overrun estimated at approximately 28% (Afana, Al Zubaidi, Abu Dabous, & Ibrahim, 2024). This phenomenon is even more pronounced in developing countries, where overruns have been reported to exceed 100% of the original project cost in some cases. A comprehensive systematic review of 69 high-impact studies published between 2000 and 2024 identified 66 interconnected cost overrun factors across construction project types, with material price fluctuations, inaccurate estimations, poor planning, and design changes consistently emerging as the dominant contributors (Abdelalim, Salem, Salem, Al-Adwani, & Tantawy, 2025). In Nepal and similar developing economies, additional contextual factors such as topographical constraints, limited access to skilled labor, and dependence on imported materials further compound the risk of cost overruns (Karn & Dahal, 2021).

2.3 Factors Influencing Building Construction Cost

The accurate identification of factors that influence building construction cost is a prerequisite for developing any reliable cost prediction model. Previous research has consistently shown that construction costs are determined by a complex and interrelated set of physical, locational, contractual, and contextual parameters (Hashemi, Ebadati, & Kaur, 2020). These factors can generally be classified into design-related parameters, site-related parameters, structural parameters, and external/market parameters. Among the design-related factors, plinth area (or gross floor area) and the number of stories is the most universally cited parameter across studies. Arafa & Alqedra (2011) identified ground floor area, typical floor area, and number of storeys as the most significant drivers of building cost in their ANN model trained on 71 building projects in Gaza. Similarly, El-Sawalhi & Shehatto, (2014) incorporated floor area and number of floors as primary inputs in a neural network model for building construction cost in the Middle East. Al-Gahtani (2024) working with 135 construction projects in Saudi Arabia, found that contract cost and contract duration together with project sector were sufficient early-stage inputs for final cost prediction with high accuracy.

Structural parameters, particularly the type of structural system and foundation type, are also well-established cost drivers in the literature. A residential houses study found structural components averaged 34.2% of total cost when designed per technical standards (Cece Suhendi, 2024). Dimitrijević, et al. (2019) demonstrated that the choice of structural system (RCC, steel, or load-bearing masonry) has a statistically significant influence on both construction cost and project duration, particularly for residential buildings. Foundation type: whether raft, isolated footing, or pile, determines a significant portion of the substructure cost and varies considerably depending on local soil conditions. The number of structural columns, which reflect structural complexity and material volume, has also been included as a relevant predictor in multiple studies (Alrasheed, 2025). Site-related factors, including the location of the building and the

`

accessibility of the construction site, further influence costs by affecting logistics, material transportation, and labor mobilization. Karn & Dahal (2021) confirmed that site accessibility and location are among the primary variables affecting building construction costs specifically in Kathmandu Valley.

External factors such as material price fluctuations, seasonal weather conditions, and inflation further complicate cost prediction. Studies in Nepal have noted that construction activity during the monsoon season is slower and more expensive due to weather-related disruptions, while the winter season affects concrete curing conditions. To control the effect of inflation over time, cost normalization using an appropriate price index is widely recommended before using historical project data for model training (Al-Gahtani, 2024).

Table 1 Factors Affecting Cost of Building Projects

Author, Date	Sample Size	Location of Study	Factors/ Key Parameters taken
Alrasheed et al. (2025)	34 building projects	Kuwait	• Building type and year • Owner and location • Excavation and concrete volume
Al-Gahtani (2024)	135 building projects	Saudi Arabia	• Contract cost and duration • Project sector (public, semi-public, private)
Sapkota et al. (2023)	72 building projects	Nepal	• Location and site access • Building type and foundation type • Floor area, floor height, and columns
Arafa & Alqedra (2011)	71 building samples	Gaza Strip	• Ground and typical floor area

`

			• Number of storeys and columns • Foundation type and number of elevators
Khanal (2025, April)	132 bridge samples	Nepal	• Project location • Structure length and width • Substructure type and construction year
MDPI (2026)	Review Study	Ethiopia	• Material price fluctuation and inflation • Material shortages • Political instability and payment delays
Deloitte (2026)	Industry Outlook	Global/US	• Rising material costs (steel/aluminum) • Tariffs and supply chain disruptions • Persistent labor shortages
Trung Hau Group (2025)	Industrial Data	Global	• Project scale and functional requirements • Geological conditions and M&E systems

`

			• Cladding materials and logistics
El-Sawalhi and Shehatto (2014)	169 building projects	Gaza Strip, Palestine	• Area of typical floor • Number of floors • Use / type of building • Type of foundation • Type of slab • Number of elevators • Type of external finishing • Presence of HVAC & false ceiling • Type of tiling • Type of electrical works • Type of mechanical/sanitary works

2.4 Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) represent a foundational paradigm in machine learning, drawing inspiration from the structure and function of biological neural systems to enable computers to learn from data rather than relying on explicitly programmed rules. At their core, ANNs are computational models composed of interconnected layers of nodes, or artificial neurons, that process information through weighted connections and activation functions, allowing the network to approximate complex, non-linear relationships within data (Goodfellow, Bengio, & Courville, 2016). Machine learning, the broader discipline within which ANNs operate, encompasses the study of algorithms that improve automatically through experience, with ANNs constituting one of its most powerful and versatile families of methods. The capacity of ANNs to learn hierarchical representations of data has driven their widespread adoption across domains such as computer vision, natural language processing, and predictive modeling, establishing them as a central subject of inquiry in both theoretical and applied research (LeCun, Bengio, & Hinton, 2015).

2.4.1 Biological Inspiration and Historical Development

The structure and operation of the human brain serve as the foundation for the idea of an artificial neural network. The biological brain uses a huge network of over 86 billion neurons, each of which is linked to thousands of other neurons by connections, known as synapses, to process information. When a neuron receives sufficient electrochemical stimulation from its inputs, it fires an electrical signal along its axon to connected neurons, enabling the brain to recognize patterns, learn from experience, and make decisions (Graupe, 2013). Artificial Neural Networks attempt to replicate this mechanism computationally using simplified mathematical representations of neurons and their interconnections.

The first formal mathematical model of a biological neuron was introduced by McCulloch and Pitts in 1943, establishing the foundational concept of a binary threshold unit that could represent logical operations (Graupe, 2013). The concept was significantly advanced by Frank Rosenblatt in 1958 with the introduction of Perceptron, which is a single-layer neural network capable of binary classification. However, the critical breakthrough that enabled the practical training of multi-layer networks came with the formalization of the backpropagation algorithm by Rumelhart, Hinton, and Williams in 1986, which provided a computationally efficient method for adjusting network weights through gradient descent (David E. Rumelhart, 1986).This development transformed ANNs from theoretical constructs into practical tools applicable to real-world prediction and classification problems, including construction cost estimation (Hashemi, Ebadati, & Kaur, 2020)

ANN has been applied in construction management since the early 1990s and the volume of ANN-based construction research has grown dramatically since 2015 (Liu, 2021). The enduring appeal of ANN in this domain lies in its ability to learn complex, non-linear mappings from input data without requiring the researcher to specify the form of the mathematical relationship in advance which is a significant advantage over regression-based methods that require explicit functional specification (Kim, An, & Kang, 2004).

2.4.2 Structure of an Artificial Neural Network

An ANN is organized into three types of layers: an input layer, one or more hidden layers, and an output layer. Each layer contains a set of nodes (neurons), and adjacent layers are connected by weighted edges. The input layer receives the raw feature values and passes them forward to the hidden layers without performing any transformation. The hidden layers are the computational core of the network, where non-linear transformations are applied to extract abstract feature representations from the input data. The output layer produces the result of ANN. Each neuron j in a hidden layer receives a weighted sum of all outputs from the previous layer and adds a bias term before applying an activation function. The mathematical operation performed by a single neuron is:

$$z_j = \Sigma_i \, (w_{ij} \times x_i) + b_j$$

$$a_j = f(z_j)$$

where x_i are the input values from the preceding layer, w_{ij} are the synaptic weights connecting input i to neuron j, b_j is the bias term for neuron j, $f(\cdot)$ is the non-linear activation function, and a_j is the output (activation) of neuron j passed to the next layer (Goodfellow, Bengio, & Courville, 2016). In matrix notation for a complete layer, this is written compactly as:

$$a = f\,(W.x + b)$$

where W is the weight matrix, x is the input vector, b is the bias vector, and f () is applied elementwise. This is the same fundamental expression used in this study’s ANN model and referenced throughout the thesis. The weights W and biases b are the

`

learnable parameters of the network their values are initialized randomly and updated iteratively during training to minimize the prediction error.

2.4.3 Activation Functions

Activation functions are a critical component of ANN design. Without them, the network would reduce to a simple linear transformation regardless of the number of layers and would be no more expressive than multiple linear regression. Non-linear activation functions enable the network to represent and learn highly complex input-output mappings. Several activation functions have been widely used in construction cost estimation research. The sigmoid function, also known as the logistic function, maps any real-valued input to the range (0, 1) according to:

$$\sigma(z) = 1 / (1 + e^{-x})$$

Its smooth, bound output makes it suitable for output layers in binary classification tasks and was historically the most common choice for hidden layers. However, the sigmoid function suffers from the vanishing gradient problem: for very large or very small inputs, the gradient approaches zero, which severely slows weight updates during backpropagation in deep networks (Goodfellow, Bengio, & Courville, 2016).

The Hyperbolic Tangent (tanh) function is a scaled version of the sigmoid, mapping inputs to the range (-1, 1):

$$\tanh(z) = (e^{x} - e^{-x} / (e^{x} + e^{-x})$$

The zero-centered output of tanh makes it preferable to sigmoid for hidden layers in many applications, as it produces stronger gradient signals during backpropagation. Several published ANN models for construction cost estimation have employed tanh as the hidden layer activation function (Arafa & Alqedra, 2011).

The Rectified Linear Unit (ReLU), defined as:

$$\text{ReLU}(z) = \max (0, z)$$

The equation has become the most widely used activation function in modern deep learning due to its computational simplicity and its ability to avoid the vanishing gradient problem for positive inputs. ReLU outputs zero for negative inputs and passes positive inputs unchanged, effectively introducing sparsity into the network's representations. In this study, ReLU was selected as the activation function for both hidden layers, consistent with contemporary best practice for regression-based ANN models trained on structured tabular data. The output layer of the model uses a linear activation function (i.e., no transformation), which is standard for regression tasks where the output is an unbounded continuous variable such as construction cost.

2.4.4 The Backpropagation Learning Algorithm

The learning process in a feedforward ANN is governed by the backpropagation algorithm, first formally described by (David E. Rumelhart, 1986), and still the foundation of virtually all modern neural network training. Backpropagation is an application of the chain rule of calculus to compute the gradient of the loss function with respect to every weight in the network, enabling those weights to be updated in the direction that reduces prediction error.

Training proceeds in two phases per iteration. In the forward pass, input data is fed through the network layer by layer using the current weight values, producing a predicted output ŷ. Loss L: the discrepancy between the prediction and the true value y, is then computed using a loss function such as Mean Squared Error (MSE). In the backward pass, the gradient of the loss with respect to each weight is computed by propagating the error signal backwards through the network using the chain rule. The weight update rule for gradient descent is:

$$w \leftarrow w - \alpha \times (\partial L/\partial w)$$

where α is the learning rate, a hyperparameter that controls the magnitude of each weight update. A learning rate that is too large causes the optimization to overshoot the minimum of the loss function and diverge; a rate that is too small causes excessively slow convergence. In this study, the Adam (Adaptive Moment Estimation) optimizer was used, which adaptively adjusts the effective learning rate for each weight parameter based on estimates of the first and second moments of the gradients, making it particularly well-suited for small and noisy datasets (Goodfellow, Bengio, & Courville, 2016). The backpropagation algorithm requires the activation functions to be differentiable, which is satisfied by sigmoid, tanh, and ReLU (for positive inputs). The iterative application of forward and backward passes over multiple epochs, complete passes through the training dataset, progressively refines the network's weights until the loss converges to a minimum or early stopping is triggered.

2.4.5 Overfitting, Regularization, and Generalization

Overfitting occurs when a model learns the specific patterns and random noise present in the training data so precisely that its performance on new, unseen data degrades which is a phenomenon that manifests as a large gap between training and test accuracy. In the context of construction cost estimation, an overfitted model would memorize the idiosyncrasies of the 56 training projects rather than learning the generalizable relationships between building parameters and cost, producing unreliable predictions for new buildings.

Several complementary strategies were employed in this study to mitigate overfitting. Dropout regularization randomly deactivates a fraction of neurons during each training batch, preventing neurons from co-adapting and forcing the network to learn more robust, distributed representations. A dropout rate of 0.20 (20%) was applied after the first hidden layer in this study's model. L2 regularization (also known as weight decay) adds a penalty term to the loss function proportional to the squared magnitude of the weights:

$$L_{regularized} = L + \lambda \Sigma w^2$$

`

where λ is the regularization strength (set to 0.005 in this study). This penalty discourages the network from assigning excessively large weights to any particular input, thereby reducing the model's tendency to memorize training data rather than generalize. Early stopping monitors the validation loss at the end of each epoch and halts training when the validation loss fails to improve for a specified number of consecutive epochs, restoring the weights from the epoch of best validation performance. This ensures that the final model reflects the point of optimal generalization rather than the point of maximum training fit.

2.4.6 Network Architecture and Hyperparameter Selection

The ability of a feedforward ANN to represent complex functions depends on its architecture, which includes the number of hidden layers, the number of neurons per layer, and the selection of activation function. There is no closed-form rule governing optimal architecture for a given dataset; it must be determined empirically through experimentation and validation. A network with too few neurons lacks the capacity to capture the complexity of the input-output relationship (underfitting), while a network with too many neurons is prone to overfitting, especially on small datasets.

A single hidden layer is theoretically sufficient to approximate any continuous function to arbitrary precision, given enough neurons a result known as the Universal Approximation Theorem (Cybenko, 1989; Hornik, 1991). In practice, however, two hidden layers often provide better empirical performance for moderate-complexity regression tasks, as they allow the network to build hierarchical feature representations: the first layer extracts low-level patterns (e.g., interactions between plinth area and storey count), while the second layer combines these into higher-level cost predictors. Most published ANN models for construction cost estimation employ one or two hidden layers (Arafa & Alqedra, 2011; Al-Gahtani, 2024; Alrasheed, 2025). Thus two hidden layers were adopted in this study based on both this precedent and the results of Bayesian hyperparameter optimization.

Bayesian optimization, the hyperparameter search strategy employed in this study, treats the model performance as a function of the hyperparameter space and uses a probabilistic surrogate model (typically a Gaussian Process) to select the most promising hyperparameter combinations to evaluate next. This approach is substantially more efficient than grid search or random search, particularly when the hyperparameter space is large and each model evaluation is computationally expensive, because it leverages information from past evaluations to direct the search toward regions of the space likely to yield better performance.

2.5 Artificial Neural Networks (ANNs) in cost estimation

ANNs have emerged as one of the most effective machine learning tools in construction cost prediction (Alrasheed, 2025). By 2030, Gartner has foreseen that up to 80% of the project management tasks will be taken over by AI through big data, natural language

`

processing, and machine learning (Audisio, 2023). ANN can play roles in prediction, optimization, classification, and decision-making in the practice of CM and has been used in CM since the early 1990s (Chao, 1998). The first mathematical model of an artificial neural network model was formulated by McCulloch and Pitts in 1943, (Graupe, 2013). ANNs are made up of interconnected artificial neurons arranged in input, hidden, and output layers, drawing inspiration from biological neural networks; each layer contains many neurons depending on the model's purpose, especially the input and output layers. The function of each neuron of the hidden layer consists of two parts: the first part is the sum of the product weights by values of neurons of the previous layer (i.e., input layer or previously hidden layer) with its bias, and the second part is to activate the result of the first part by a specific function (tangent hyperbolic or sigmoid). By identifying intricate patterns in past data, these networks are trained to improve prediction models. According to (Hashemi, Ebadati, & Kaur, 2020), artificial neural networks (ANNs) have been extensively utilized in construction cost estimation research over the past 35 years. Previous studies consistently demonstrate that Artificial Neural Networks (ANNs) offer a more accurate and reliable approach to cost estimation compared to traditional regression models, especially in the early stages of a project where information is limited. Machine learning models achieve an average accuracy of 75–80%, providing strong performance, making them highly effective for handling complex, nonlinear relationships and large datasets. (Shamim, Hamid, Nyamasvisva, & Rafi, 2025).

A critical aspect of developing ANN models for cost prediction is the selection of architecture and hyperparameters, including the number of hidden layers, neurons per layer, activation functions, learning rate, batch size, and number of training epochs. There is no universal rule governing optimal architecture; rather, it is determined empirically through experimentation and validation on the specific dataset (Goodfellow, Bengio, & Courville, 2016). Most published ANN models for construction cost estimation employ a single hidden layer, which has been found sufficient for capturing the non-linear relationships present in building cost data when the dataset size is moderate (Arafa & Alqedra, 2011; Al-Gahtani, 2024). Commonly used activation functions include the sigmoid function and the Rectified Linear Unit (ReLU), both of which introduce the non-linearity necessary for the network to go beyond simple linear regression. The 80:20 train-test split ratio is the most widely adopted approach for model validation in construction cost studies, ensuring that the model is evaluated on data it has never encountered during training (Khanal, 2025, April).

Figure. 2 shows a condensed version of a feedforward ANN model that uses activation functions and weighted connections to enable learning and generalization from prior

`

datasets.

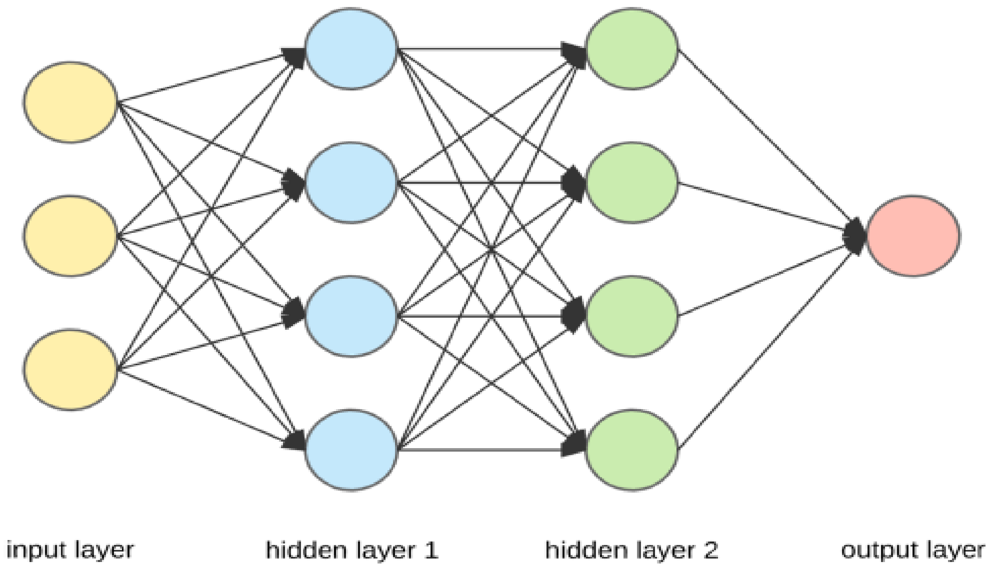

Figure 2 ANN diagrammatic representation

ANNs are broadly classified into two types: feedforward networks and recurrent networks. For tasks such as cost estimation using past project data - a feedforward architecture with backpropagation is typically employed (Acharya & Karki, 2022, October). In a feedforward network, neurons are fully interconnected across layers, processing data in a unidirectional flow. This structure includes an input vector (x), a weight matrix (W), a bias vector (b), and an output vector (y), expressed mathematically as:

$$y = f\,(W \cdot x + b)$$

where, f () denotes a non-linear activation function, such as sigmoid or ReLU, which enables the network to capture complex patterns.

2.6 Global Context of ANN

The use of Artificial Neural Network has seen a dramatic increase in the sector of construction management since 2015, according to research conducted by (Liu, 2021). Cost and performance measurement are particularly common areas of application of ANN in recent times, with cost estimation of construction projects being the most widely discussed topic among others. For construction cost estimation, ANN is a representative method for early construction cost estimation by identifying cost influencing factors and establishing a prediction model based on historical data (Xu, et al., 2022). Juszczyk, Zima, and Lelek presented an original approach of building construction cost predictive models based on ensembles of some MLPNNs (Liu, 2021). Rafiei and Adeli used advanced machine learning concepts to create innovative

`

construction cost estimation models, including an unsupervised deep Boltzmann machine learning approach and a soft-max layer three-layer BPNN (Liu, 2021).

A scientometric review conducted by Xu, et al. (2022) demonstrated that countries like China, Taiwan and the USA are leading in the number of research contributing to the application of ANN in the field of Construction Management. The review suggests that this field will garner more interest and research in the future. It validates the fact that ANN has been successfully applied to the intelligent solution of problems such as prediction, success, cost, productivity, risk and safety throughout the whole life cycle of construction projects without too much manual intervention (Xu, et al., 2022).

2.7 ANN use context in Nepal

Traditional cost estimation methods in the construction sector have long been recognized as methodologies riddled with uncertainty, requiring significant improvements in forecast accuracy (Acharya & Karki, 2022, October). Conventional estimation approaches, such as regression models and expert judgment, struggle to capture the non-linear relationships between cost-influencing factors, leading to inconsistent and unreliable forecasts (Khanal, 2025, April). Quantity Rate Analysis is the primary conventional method for estimating costs that is commonly utilized (Veliyampatt). Therefore, a need for a better cost forecasting method is apparent in the construction industry of Nepal. The use of ANN forecasting model enables planners and decision makers to make a more informed decision regarding the allocation of limited capital budget on projects. Despite the growing popularity of machine learning in various industries, its application to the construction sector in Nepal has been relatively limited (Sapkota, Karki, Pokharel, & Dhital, 2023). The use of ANN based forecasting model can become a powerful decision-making tool at the project conception phase to the investors, project managers and all related stakeholders, thus helping supplement the economic feasibility of building construction projects.

`

CHAPTER 3: METHODOLOGY

This chapter offers a thorough summary of the research techniques used to accomplish the goals of the study. This chapter covers research methodology, research design, study area, population size, sample size computation, statistical tools and tests, data collection methods, data analysis, and research matrix. The steps taken to implement the research design are shown graphically in Figure 3.

First and foremost, appropriate research and literature on building construction costs were examined to comprehend a thorough overview of the study topic and to collect thoughts regarding earlier research in the relevant sector. The elements influencing building construction prices in the Kathmandu Valley were to be investigated due to the chosen study questions. Technical papers, national and international journals, and other resources were examined as part of the literature to cover a vast amount of information about the subject of this study.

3.1 Research Methodology

The research methodology follows a structured approach. Developing the research concept was the first stage before beginning the actual research procedure. This was done by identifying the challenges that made the research necessary. The scope and goal of the study were established by the research objectives. The research question served as the foundation for the study objective The next stage was the development of a conceptual framework, which functioned as a road map outlining the techniques and protocols for gathering and evaluating data in order to get the best results that could solve the research challenge.

This study employs a quantitative research approach to develop an Artificial Neural Network (ANN) model for early-stage cost estimation of building construction projects in Kathmandu valley. The methodology is structured to address the research objectives: identifying factors influencing building construction costs and creating an accurate ANN-based cost prediction model. The process involves literature review, data collection, data preprocessing, and model development.

3.2 Research Design

The strategy for carrying out the investigation is called the research design. It is a technique designed to answer research questions It entails choosing the sample techniques, characterizing the population, and identifying the key variables (factors) in order to build a study path. In order to collect quantitative secondary data for the study, a variety of organizations and professionals involved in building construction in the

Kathmandu Valley were contacted. The data collected was preprocessed, validated through experts, and the parameters needed for our study were extracted.

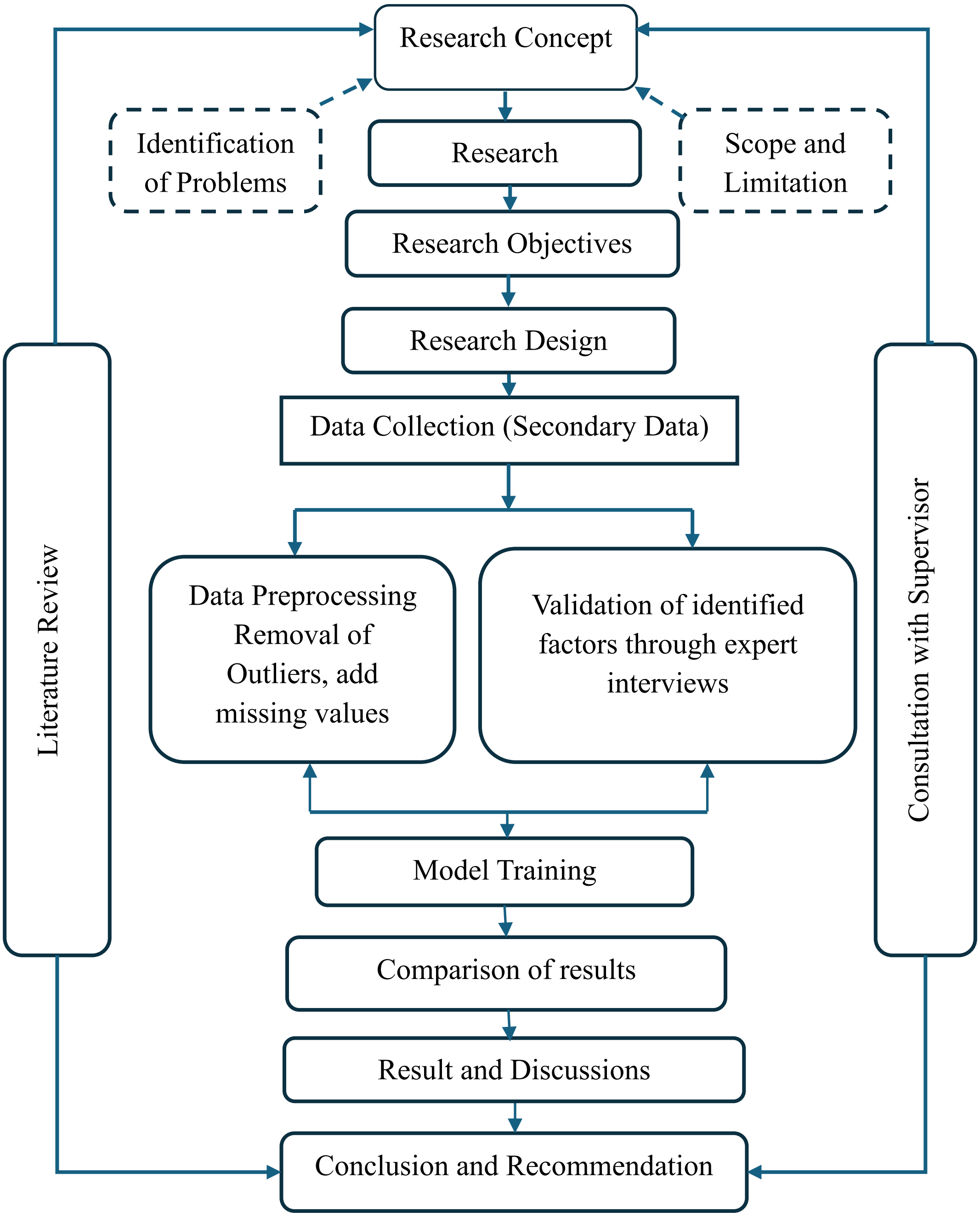

Figure 3 Research Design

3.3 Research framework

The first specific objective of this research is to systematically identify the primary factors that influence the final construction cost of residential buildings. This objective is foundational to the entire study because the quality and relevance of the input variables fed into the ANN model directly determines the model's predictive accuracy and practical utility. Without a rigorous, evidence-based identification of cost-influencing factors, any machine learning model built upon them risks being unreliable or context-irrelevant. This objective is therefore addressed in two complementary phases: an extensive review of existing literature and a structured expert validation exercise.

In the first phase, a comprehensive review of national and international literature on building construction cost estimation is conducted. Published research papers, technical reports, and journal articles spanning a wide range of geographies and building types are examined to compile an initial pool of parameters consistently found to influence construction costs. From this review, two categories of variables emerge as recurring and significant cost drivers: numerical parameters such as the number of storeys, plinth area, floor height, site accessibility coefficient, and number of structural columns; and categorical parameters such as building classification, foundation type, structural system commencement, and district location within Kathmandu valley. This step of the research framework ensures that the model's input space is grounded in a well-established academic and professional body of knowledge rather than arbitrary assumptions, and the full list of initially identified variables is documented in Tables 4 and 5 of the Results and Discussions chapter.

In the second phase, the initially identified factors are contextualized to the specific construction environment of Kathmandu Valley through structured expert interviews. A questionnaire containing the parameters identified from literature is designed and administered through KoboToolbox to five experienced professionals drawn from the Ministry of Urban Development (MOUD) the Department of Urban Development and Building Construction (DUDBC), and private contracting firms with direct involvement in residential construction projects. The expert validation exercise serves the critical purpose of filtering out variables that, while theoretically relevant in other geographic contexts, do not meaningfully contribute to cost variation in Kathmandu Valley's residential construction setting.

The second specific objective is to investigate and establish the nature and strength of the relationships between the identified input variables and the final construction cost, as well as the inter-relationships among the input variables themselves. This objective is critical because understanding how variables relate to one another and to the target output determines the suitability of an ANN approach over simpler linear models. If the relationships were straightforward and strictly linear, traditional regression methods would suffice. However, this objective is pursued through rigorous exploratory data analysis and statistical examination of the collected dataset to demonstrate that the cost dynamics of residential buildings in Kathmandu Valley are inherently multi-variate and non-linear, thus justifying the adoption of an Artificial Neural Network as the modelling strategy.

This objective is addressed primarily through the data preprocessing and exploratory data analysis stages of the research. After historical data from 76 completed residential building projects is collected from construction firms and consultants operating across the three districts of Kathmandu Valley, the dataset undergoes a sequence of preprocessing operations designed not only to prepare it for model training but also to reveal the distributional and relational patterns inherent in the data. These operations include the removal of outliers using asymmetric percentile trimming, Wholesale Price Index (WPI)-based cost normalization to a common base year of 2025/26 to control for inflation effects across different construction years, min-max scaling of numerical variables to the range of [0, 1] to eliminate scale dominance, and one-hot encoding of categorical variables to transform location and foundation type into numerical binary vectors suitable for ANN processing. Following preprocessing, a scatter matrix and a correlation heatmap are generated to visually and statistically characterize the relationships within the dataset.

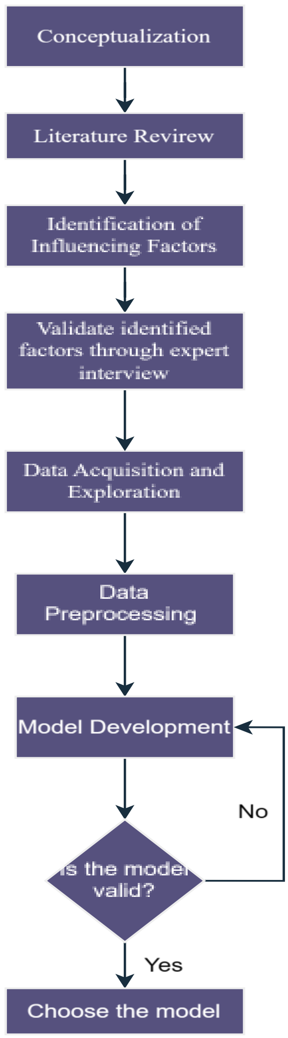

Figure 4 Research Methodology Framework

`

The third and central specific objective of this research is to design, train, and validate an Artificial Neural Network model that can accurately and efficiently predict the final construction cost of residential buildings in Kathmandu Valley using only early-stage project parameters as inputs. This objective represents the culmination of the entire research effort and is addressed through the model development phase, which encompasses architectural design, hyperparameter tuning, training, validation, and performance evaluation. The objective is considered achieved when the trained model demonstrates statistically acceptable prediction accuracy on an independent test dataset that it has not encountered during training, thereby confirming its generalization capability for use in real-world early-stage project cost estimation. The ANN model is implemented using the TensorFlow 2.x, scikit-learn libararies and Keras framework in Python and is built as a feedforward, fully connected neural network trained using the backpropagation algorithm. The preprocessed dataset of 76 building projects after removal of outliers is split into a training set of 85% (56 samples) and a test set of 15% (11 samples), ensuring that model evaluation is conducted on unseen data to provide an unbiased assessment of predictive performance.

3.4 Study Area

The scope of our study is limited to buildings constructed in Kathmandu valley, which include three districts, namely, Kathmandu, Bhaktapur, and Lalitpur. These districts have been growing with huge number of residential buildings being built every year, which raises the question about the accuracy of cost estimated at the feasibility study stage of project.

Population, sampling and sample size

Population of the study was sampled from an infinite number of building data constructed in the valley in recent years. The districts within the valley, namely Kathmandu, Bhaktapur and Lalitpur were sampled for data through the method of convenience sampling.

Calculation of sample size

Cochran (1963) developed a formula to determine the required sample size for proportions in infinite population. It is calculated using

$$n_o = z^2pq/e^2$$

Where,

n_o = Sample size

Z = z-score (1.645 for 90% confidence Interval)

P = Estimated proportion (0.5 for maximum variability)

q = 1-p

`

e = Margin of error (10% i.e. 0.1)

So, $n = \frac{1.645^2 \times 0.5 \times (1-0.5)}{0.1^2}$

$= 67.65 \approx$ **68 samples**

Convenient sampling was employed to collect the required data

3.5 Method of Data Collection

Identification of Influencing Factors

To determine the factors affecting building construction costs an initial set of input variables was identified through an extensive review of prior studies on buildings cost estimation. This would provide valuable insights into common parameters which were adopted to Kathmandu valley's context through the guidance of experts.

Validation through Experts

The identified factors were validated through interviews carried out with experts who have substantial experience in the field of building construction. A total of 5 experts were provided with questionnaires containing parameters identified through literature review. The parameters thus validated through experts were chosen as the major factors influencing building construction costs in Kathmandu valley.

The first phase of the data collection process involved gathering and exploring historical data (secondary data) from various completed building construction projects in Kathmandu valley which included the districts of Kathmandu, Bhaktapur and Lalitpur. Thus, the data required for this study was sourced from private construction companies and consultants that have been undertaking residential building construction projects in Kathmandu valley. Drawings and final costs of buildings constructed within the valley were collected for further analysis and preprocessing. A residential houses study found structural components averaged 34.2% of total cost when designed per technical standards (Cece Suhendi, 2024). Thus, structural components were chosen as major factors affecting residential building costs excluding other parameters such as HVAC systems, electrical and sanitary works and other finishing/decorative works. Structural data and final costs of 76 (seventy-six) residential buildings were collected.

Table 2 Numerical Values Range of Sampled Building

Variable	Minimum Value	Maximum Value	Average Value
No. of Storeys	2	5	3.30
Plinth Area	55.00	127.18	80.61
No. of Columns	9	201.6	35.46
Adjusted Cost	5404321	19500000	11652738.54

`

3.6 Data Analysis

Data Preprocessing

A crucial stage in machine learning is data preprocessing, which entails converting unprocessed data into a clear, uniform format that is appropriate for model training and analysis. Raw data from construction projects often contain inconsistencies, missing values, varying scales, or categorical entries that can hinder an ANN's performance if not addressed (Theobald, 2021). Preprocessing enhances data quality, improves model accuracy, and ensures compatibility with ANN algorithms by normalizing inputs, handling missing data, and encoding variables appropriately. Data preprocessing transforms raw data into a format suitable for ANN modeling. To reduce the influence of extreme values that could distort model training, outlier removal was applied: projects in the bottom 10th percentile and top 1st percentile of the cost distribution were excluded. This asymmetric trimming was intentional as, the lower tail was removed more aggressively to eliminate data entry anomalies, while only the most extreme upper values were excluded to preserve genuine high-cost projects in the dataset.

3.7 Model Development

An ANN model was designed and implemented to predict building construction costs. This process involves exploring different ANN architectures, including the number of layers, neurons per layer, and activation functions, to identify the most effective configuration. The model was trained using the preprocessed dataset, allowing it to learn the complex, non-linear relationships between the input parameters and the final construction costs. The model was developed on Python programming language through the use of various libraries for data pre-processing and machine learning. The model development focuses on building and fine-tuning a neural network with optimal architecture. The feedforward neural network with backpropagation was used to train the model where data flows in one direction from input through hidden layers and to the output layer while the backpropagation algorithm is used to minimize the error between predicted and actual project duration by adjusting the network's weights and biases iteratively (David E. Rumelhart, 1986).

The dataset was entered in CSV format which was split into an 85-15 split ratio so that the model was trained on one subset of the data and evaluated on an unseen subset. The hyperparameters like epochs, batch size, and learning rate are treated as variables and various combinations of these parameters form various ANN architectures.

Model Evaluation Metrics

The model's performance was assessed using metrics such as MAPE and R^2 which provide insights into its accuracy and predictive capability.

1. R-Squared (Coefficient of Determination):

The R^2 value measures how much of the variability in the dependent variable can be

`

explained by the independent variables in the model. It is computed using the formula:

$$R^2 = 1 - SSR/TSS$$

Where,

SSR (Sum of Squared Residuals) quantifies the difference between the actual values and the predicted values produced by the model.

TSS (Total Sum of Squares) represents the overall variation present in the actual values of the target variable.

2. Mean Absolute Percentage Error (MAPE):

MAPE is a widely used metric for evaluating the accuracy of forecasting and regression models. It expresses prediction errors as a percentage, making it intuitive and easy to interpret across different scales.

$$MAPE = 1/n \left[\sum_{i=1}^{n} (A_i - F_i) / A_i\right]$$

where:

A_i = Actual value

F_i = Forecasted value

n = Number of observations

MAPE values below 10% are classified as high accuracy, 10-20% as good accuracy, 20-50% as reasonable accuracy, and above 50% as inaccurate forecasting. (Lewis, 1982)

3.8 Research Matrix

Table 3 Research Matrix

Objectives	Research Questions	Methods	Data Sources	Expected Outcomes
To identify the parameters affecting the construction cost of residential buildings	What are the parameters identified in previous studies?	Literature Review	Research papers, articles, Journals	List of key parameters influencing building costs
	What are the major parameters affecting building costs?	Expert Interview	Experts from Contracting companies, Government entities	Identification of factors affecting building construction costs in Kathmandu valley
To identify the relationship between variables that affect the construction costs of residential buildings in Kathmandu valley.	What is the relationship between identified variables that affect the construction costs of buildings in Kathmandu valley.	Descriptive Statistics, Scatter Plots, Heatmaps	Historical Building Data collected through construction firms and consultants	Visual Representation of relationship between the variables.
To develop an ANN model that predicts the final construction cost of residential buildings in Kathmandu valley.	What are the optimal architecture and parameters for the ANN model?	Experiment with different architecture	Historical Building Construction costs	Optimal ANN model architecture
		Hyper Parameter Tuning		
	How accurate is the ANN model in predicting building costs compared to traditional methods	Performance metrics (MAPE and R^2 values)	Actual construction project costs and predicted output	Accuracy of models in terms of MAPE
	How can the ANN model be validated for reliability?	K-fold cross validation	Training and Test Data set	Robust and reliable ANN model with confirmed predictive capability.

CHAPTER 4: RESULTS AND DISCUSSIONS

4.1 Identification of the parameters affecting the construction cost of buildings

The final input variables were determined first, through a comprehensive literature review and then validation through expert consultation. From the wide range of factors influencing building construction cost identified in the literature, the most critical parameters were shortlisted for validation in the context of Nepal. Most studies and industry breakdowns suggest structural components commonly account for roughly one-third of total building construction cost. The factors identified through literature review were further classified as numerical parameters and categorical variables and listed below in Table 4 and 5:

Table 4 Numerical Input Variables

S.N.	Variables	Units
1	Number of Storeys	Number
2	Plinth Area	Sq. m.
3	Floor Height	Meter
4	Accessibility of Site	Number
5	Number of Columns	Number

Table 5 Categorical Input Variables

S.N.	Variables	Categories
1	Classification of Building	Residential, Hospital, Administration, Halls, Quarters
2	Season of construction commencement	Summer, Autumn, Monsoon, Winter
3	Location of Building	Kathmandu, Bhaktapur, Lalitpur
4	Foundation Type	Raft, Isolated, Pile,
5	Structural System	R.C.C., Steel, Load Bearing,

Structured interviews were then conducted with experienced professionals from the Ministry of Urban Development (MOUD), the Department of Urban Development and Building Construction (DUDBC), and professionals with specific expertise in building construction, using a questionnaire shared through KoboToolbox, provided in Appendix A. The insights from the interview were analyzed and documented in Appendix B. Based on the expert feedback, the key factors significantly affecting building construction costs in Kathmandu valley were identified. The buildings in the scope of this study are generally built with similar floor height thus limiting the variability of data to train ANN model and they are highly accessible for transportation of construction materials throughout the valley. Thus, Floor Height and Accessibility of Site were excluded from further study. Regarding categorical input variables, expert interview results suggested that season of commencement of building construction had

`

no impact on cost of construction. Classification of buildings and structural system parameters were also excluded from further study because the study scope was limited to residential buildings which were mostly constructed with RCC structures.

Table 6 Final Numerical Input Variables

S.N.	Variables	Units
1	Number of Storeys	Number
2	Plinth Area	Sq. m.
3	Number of Columns	Number

Table 7 Final Categorical Input Variables

S.N.	Variables	Categories
1	Location of Building	Kathmandu, Bhaktapur, Lalitpur
2	Foundation Type	Raft, Isolated, Pile,

4.2 Identification of the relationship between variables that affect the construction costs of buildings in Kathmandu valley.

The data distribution of various numerical data is as follows:

Table 8 Descriptive Statistics of Numerical Variables

Descriptive Statistics	**No. of Columns**	**Plinth Area**	**No. of Storey s**	**Adjusted Cost**
Mean	9.99	80.61	3.30	11652738.54
Standard Error	0.16	1.82	0.10	334799.2647
Median	9.00	81.00	3.00	11326836
Mode	9.00	91.23	3.00	12398148
Standard Deviation	1.44	15.87	0.88	2918712.323
Sample Variance	2.07	251.72	0.77	8518881621777.52
Kurtosis	-1.23	-0.20	-0.50	0.586205798
Skewness	0.72	0.37	0.33	0.623794507
Range	5.00	72.18	3.00	14095679
Minimum	8.00	55.00	2.00	5404321
Maximum	13.00	127.18	5.00	19500000
Sum	759.00	6126.60	251.00	885608129
Count	76.00			

`

The data distribution of various categorical variables is as follows:

Table 9 Location Count

Location	Count	Percentage
Kathmandu	48	63.16
Bhaktapur	17	22.37
Lalitpur	11	14.47

Table 10 Foundation Type Count

Foundation Type	Count	Percentage
Isolated	57	75
Mat	19	25

Table 11 Foundation Type vs Location (Cross-Tabulation)

Foundation Type	Bhaktapur	Kathmandu	Lalitpur
Isolated	13	35	9
Mat	4	13	2

Data Preprocessing

Cost Adjustment

Actual construction costs are standardized to a base year using the Wholesale Price Index (WPI) from Nepal Rastra Bank's National Statistics to adjust for inflation. All collected data were standardized to base year cost 2025/26 to ensure that the final construction costs were compatible for comparison.

It is calculated as:

$$\text{Adjusted Cost}_{2025/26} = (\text{Actual Cost in the year} / \text{WPI}_{\text{year}}) \times \text{WPI}_{2025/26}$$

Normalization/Scaling

It is done to adjust numerical data to a common scale [0,1] to prevent features with larger ranges (e.g., plinth area in square meters) from dominating smaller ones (e.g. number of storeys) during ANN training (Goodfellow, Bengio, & Courville, 2016).

$$n = (X - X_{min}) / (X_{max} - X_{min})$$

Categorical Encoding

Foundation type (isolated or mat) and location (Kathmandu, Bhaktapur, Lalitpur) were one-hot encoded to convert categorical data into binary numerical inputs, a common ANN preprocessing technique. One-hot encoding transforms each category into a binary vector. Each category is represented by a new column (feature), and for any given record, only one of those columns will have a value of 1, while the others are 0.

This ensures that all categories are treated equally without implying any sort of ranking or magnitude.

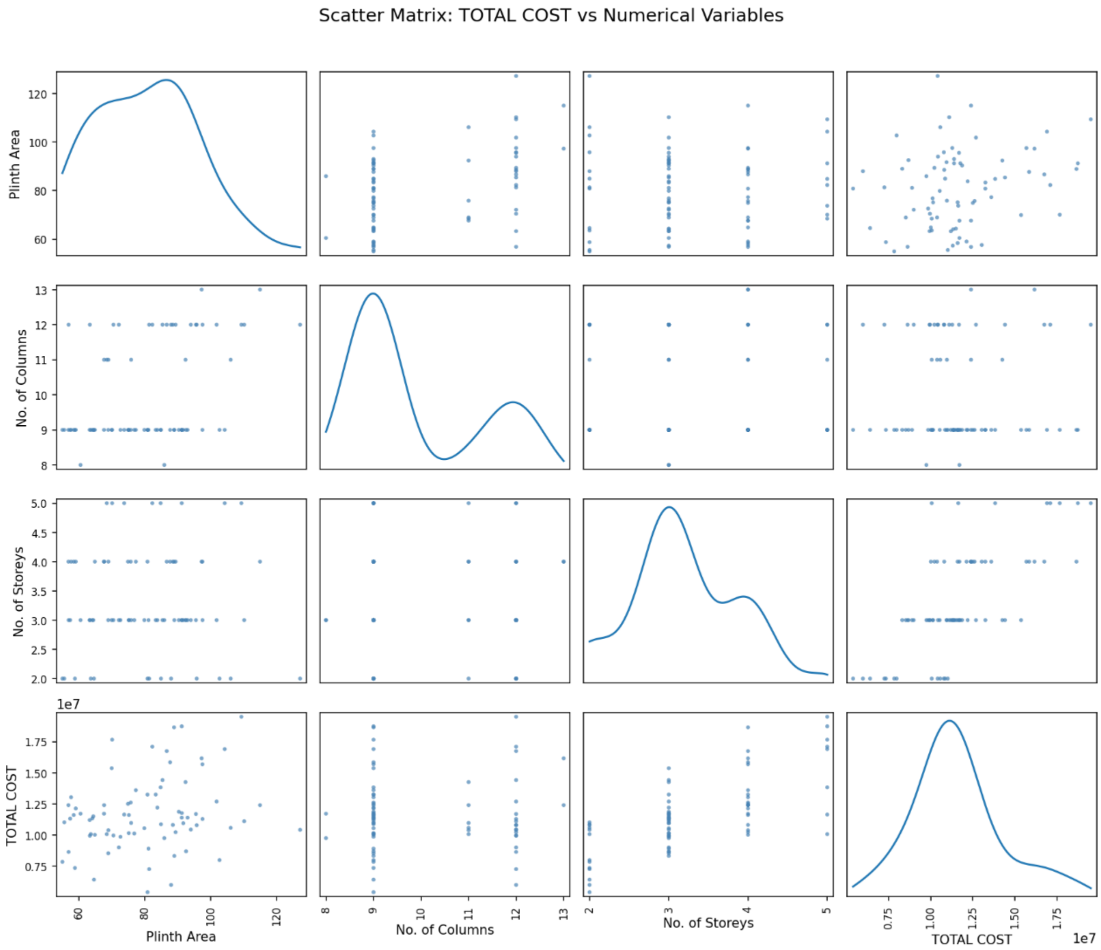

Figure 5 Scatter plot of variables

The figure above (Figure 5) shows a scatter matrix showing the relationships between four variables: plinth area, no. of storeys, product of area and cost (M NPR), across three locations: Kathmandu, Bhaktapur, and Lalitpur. According to the results, the interaction of plinth area and number of storeys shows the strongest and most consistent positive relationship with construction cost across all three locations, confirming it as the most valuable predictor in the dataset. Plinth area has a moderate positive relationship with cost but with wide scatter, meaning it alone is insufficient as a predictor. Storey count is largely concentrated at 3-storeys across all districts with some 4-storey buildings, and while higher storeys generally mean higher cost, the wide cost range within each storey count confirms that other variables are needed to explain cost differences.

Kathmandu dominates the dataset in sample size, has the widest range of building sizes, and has the highest cost variability, extending up to NPR 18 - 20 million. Bhaktapur projects are more uniform and consistently lower cost. Lalitpur has a small sample but shows a tendency toward higher cost relative to its building size. Across all panels, the three districts do not form clearly separate clusters, meaning location alone is not a strong cost differentiator and it contributes in combination with structural and design variables. The overarching implication of these plots is that construction cost in Kathmandu valley is genuinely multi-variate and non-linear, with no single variable cleanly predicting cost. This is exactly the condition under which ANN outperforms regression.

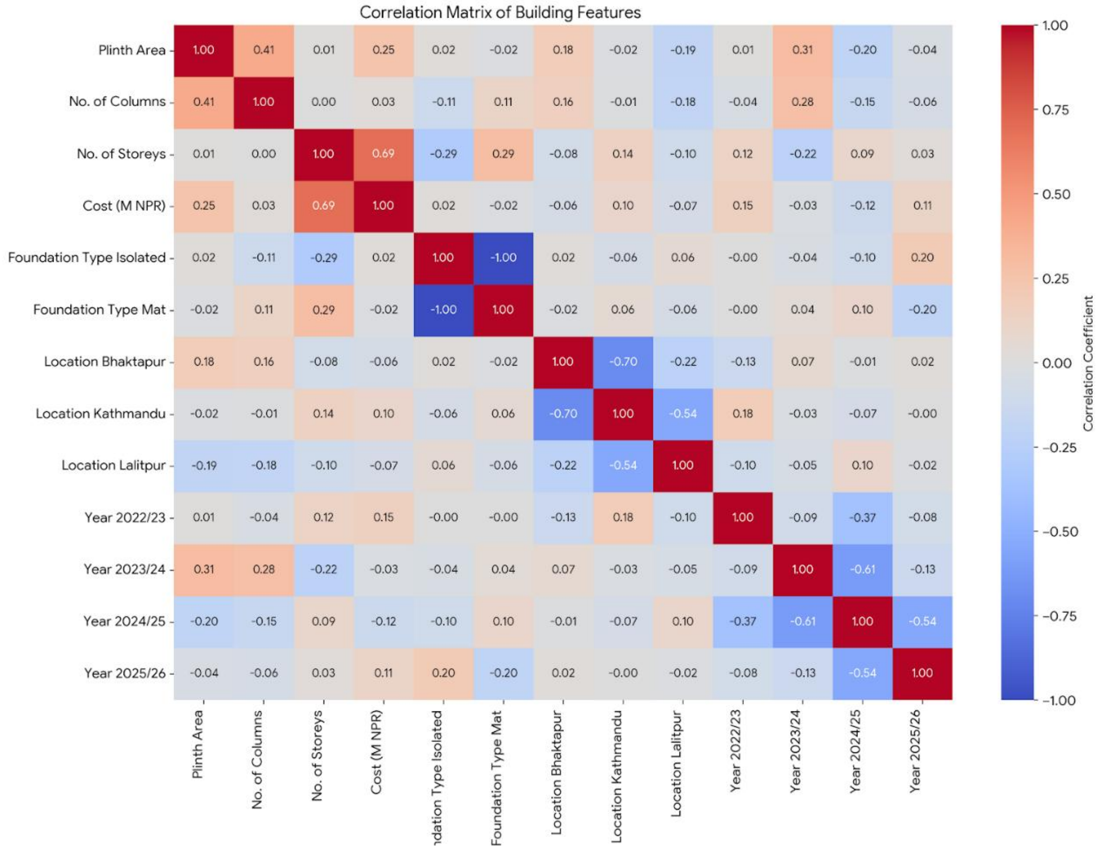

Figure 6 Correlation Heat Map of Variables

The heatmap shown in Figure 6 provides a statistical overview of how various building parameters relate to the final construction cost and to each other. The most significant relationship in the dataset is between No. of Storeys and Cost (Million NPR), which shows a strong positive correlation of 0.69. This indicates that as the height of the building increases, the total cost rises substantially. Plinth area shows a moderate positive correlation with cost (0.25), reinforcing that the horizontal footprint is a key factor, though secondary to the vertical scale (storeys). Location Kathmandu has a slight positive correlation with cost (0.10), while Lalitpur (-0.07) and Bhaktapur (-0.06) show slight negative correlations, suggesting Kathmandu-based projects in the dataset might lean toward higher costs.

`

4.3 Development of an ANN model that predicts the final construction cost of residential buildings in Kathmandu valley.

This study aimed at developing an Artificial Neural Network (ANN) model for early cost estimation of residential building projects in Kathmandu valley, utilizing 76 datasets from building construction firms and consultants. The methodology involved identifying cost-influencing factors, preprocessing data, and training an ANN model with five input features: plinth area, number of columns, number of storeys, location of building and foundation type. The ANN model was trained on 85% of the dataset (56 samples) and validated on the remaining 15% (11 samples), adhering to standard machine learning best practices to ensure unbiased evaluation. Data preprocessing standardized costs to the 2025/26 base year using the Wholesale Price Index (WPI) developed by NRB and converted categorical variables to numerical via one-hot encoding. The ANN model was implemented using TensorFlow 2.x / Keras in Python programming language. The ANN network developed is a feedforward fully connected (dense) architecture with two hidden layers. The model architecture was progressively optimized by tuning hyperparameters such as the number of hidden layers, epochs, learning rate, number of neurons in the hidden layer, drop rate through the range as shown in Table 12. This was achieved using the Bayesian optimization technique, which selects the next set of parameters to test by leveraging past performance. It balances exploration of new options with exploitation of promising ones, enabling efficient and effective identification of the optimal configuration.

Table 12 ANN Architecture and Hyperparameter Specification

Parameter	**Value**	**Role**
Input Layer Dimension	Determined by preprocessor output, i.e., 10	Captures all numerical polynomial interactions and categorical encodings
Hidden Layer	2 hidden layers with 128 and 64 neurons, ReLU activation	Primary feature learning layer; ReLU introduces non-linearity and avoids vanishing gradient
Dropout (after Layer 1)	Rate = 0.20 (20%)	Randomly deactivates 20% of neurons per batch to prevent co-adaptation and reduce overfitting
L2 Regularization	λ = 0.005 (both hidden layers)	Penalizes large weight magnitudes to discourage

		memorization of training noise
Output Layer	1 neuron, linear activation	Single continuous output; linear activation used for unbounded regression target
Optimizer	Adam (learning rate = 0.001)	Adaptive learning rate optimizer; well-suited for noisy gradients in small datasets
Loss Function	Mean Squared Error (MSE)	Standard regression loss; penalizes large prediction errors more heavily than MAE
Batch Size	8	Small batch size suitable for limited dataset; provides noisy gradient estimates that assist generalization
Maximum Epochs	500	Upper bound; actual training terminated early by Early Stopping callback
Early Stopping	Patience = 50 epochs; monitors value loss; restores best weights	Stops training when validation loss ceases to improve; prevents overfitting beyond the optimal point
Learning Rate Reduction	Reduce LR On Plateau: factor = 0.5, patience = 15, min. learning rate = 1×10^{-5}	Halves learning rate when validation loss plateaus for 15 epochs; allows finer convergence near optima
Random Seed	NumPy:42; TensorFlow: 42	Ensures full reproducibility of weight initialization and training dynamics

The optimum result was obtained when number of hidden layers was 2, and the neural architecture was 10-128-64-1 with learning rate being 0.001, drop rate was 0.2, batch size was 8 and number of epochs was 500 with the following evaluation metrics

`

Table 13 Evaluation Metrics Data ANN model No. 1

Metric	Training Data	Testing Data
R^2 Score	0.9019	0.6787
MAPE (%)	4.97%	7.32%

This model achieved a Mean Absolute Percentage Error (MAPE) of 7.32% and an R^2 value of 0.6787 on the validation dataset as seen in Table 13. A MAPE of 7.32% indicates that, on average, the model's cost predictions were within approximately 7% of the actual values, while the R^2 value suggests that 67.87% of the variation in residential building construction costs could be explained by the input variables. The drop of R^2 in testing data indicates mild overfitting: the model has learned the training data somewhat too specifically and does not generalize as strongly to new projects as a consequence of limited dataset size. These results demonstrate that the model has a reasonably good predictive ability.

The ANN model was again trained on 80:20 and 70:30 split adhering to standard machine learning best practices to ensure unbiased evaluation and the R^2, MAPE values for training and test data were as follows:

Table 14 Evaluation Metrics Data ANN model No. 2 and 3

Metric	Training Data	Testing Data
80:20 split		
R^2 Score	0.4393	0.7362
MAPE (%)	12.37%	6.00%
70:30 split		
R^2 Score	0.7398	0.3595
MAPE (%)	7.86%	8.95%

The data in Table 14 shows that ANN model trained and tested with 80:20 split performs better on testing data (R^2 = 0.74, MAPE = 6%), but poorly on training (R^2 = 0.44), suggesting underfitting. The 70:30 split shows an opposite trend demonstrating strong training performance (R^2 = 0.74, MAPE = 7.86%) but weak testing (R^2 = 0.36), indicating overfitting. Neither split achieves consistently good performance across both training and testing, highlighting a generalization issue in the models.

`

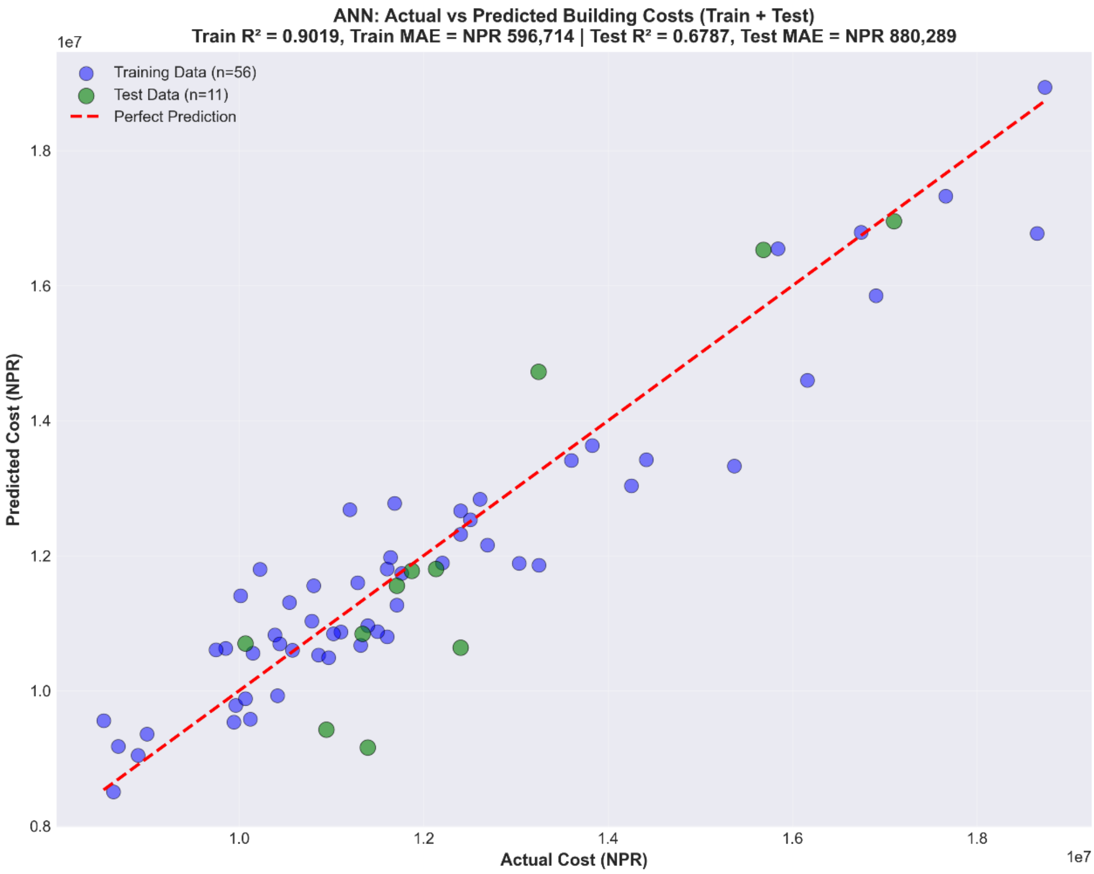

Figure 7 Actual Vs Predicted Cost for Training and Test Data

The scatter plot in Figure 7 illustrates the model's predictive performance by comparing actual versus predicted cost at completion values for both training and test datasets. Each dot represents one building project where blue and green dots represent the training and test data predictions respectively. Its x-position is the actual construction cost, and its y-position is what the ANN predicted. Points lying exactly on the red dashed line represent perfect predictions. Points above the line mean the model over-predicted; points below mean it under-predicted. The model is most reliable for average complexity projects in the NPR 9-13 million range, which represents the majority of residential construction in the valley. For high-cost outlier projects, the model should be used with caution and supplemented with professional judgment.

Model Validation

The best combination from the above model was found to be the first model with neuron architecture 10-128-64-1 based on the MAPE and R^2 of the test set. However, to validate that the model is generalizable, 10-fold cross-validation was carried out and the MAPE and R^2 values of Training and Validation sets are tabulated in Table 15:

Table 15 R^2 and MAPE values for each fold

Model	R^2 Score	MAPE (%)
ANN model 1	0.6044	7.22
ANN model 2	0.5515	7.00
ANN model 3	0.5186	7.53
ANN model 4	0.4619	8.22
ANN model 5	0.6908	5.90
ANN model 6	0.4567	7.43
ANN model 7	0.3646	7.83
ANN model 8	0.5365	8.87
ANN model 9	0.5048	7.53
Average	**0.5211**	**7.50**

The Mean Absolute Percentage Error (MAPE) value can be visualized through the bar graph as shown in Figure 8, which shows that ANN model 5 has the least value of MAPE i.e., 5.90%.

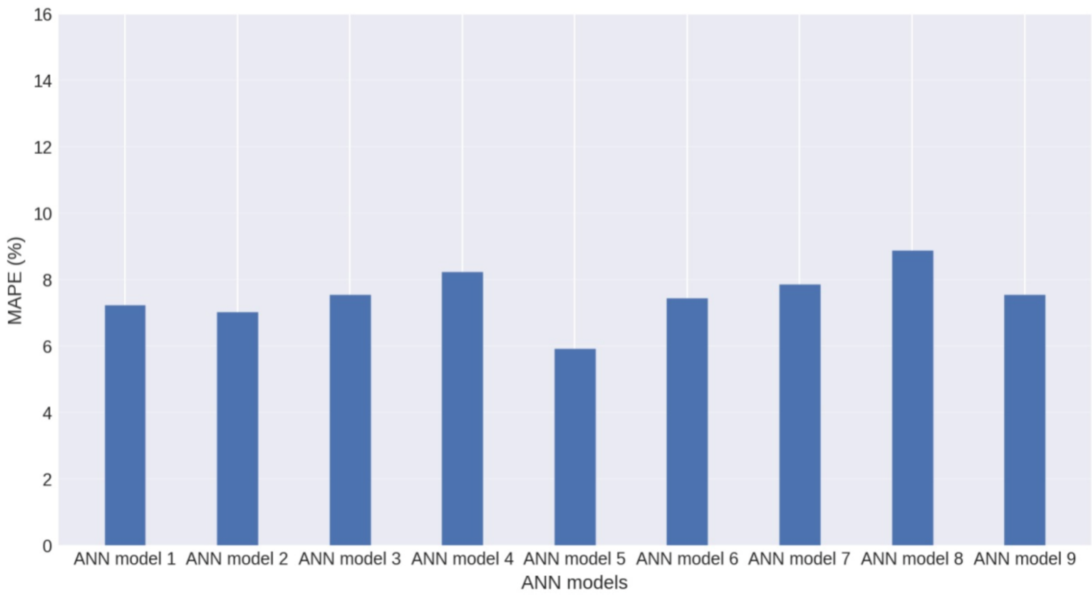

Figure 8 Bar Chart showing MAPE of each ANN model

Based on the optimal architecture obtained from the above results, i.e., ANN model No. 5, with a MAPE of 5.90% and an average R^2 value 0.6908, we can now predict the duration of all the datasets which will include training, validation and test sets. Previous study in Saudi Arabia by Al-Gahtani (2024) for final construction cost prediction of buildings at an early stage resulted MAPE's mean, maximum, and minimum values as 18.7%, 20.15%, and 16.78%, respectively from first analysis and it ranged from 5.77% to 15.8%, with a mean of 8.7% in the second analysis. Another study by Alrasheed (2025) on ANN based cost estimation conducted on public construction projects in Kuwait resulted in a mean absolute percentage error (MAPE) of 0.72%, indicating a prediction accuracy of 99.28 % on a data sample of 28 publicly funded buildings. Finally, Artificial Neural Networks for cost estimation of road projects in Nepal by Acharya & Karki (2022, October) resulted a model with architecture 14-6-9-9-1 producing the best result with average validation and training MAPE of 13.90% and 12.24%

To visualize the accuracy of the trained model's prediction, a scatter plot is generated in which x-axis represents actual duration while y-axis represents the predicted duration. The proximity of data to the line of perfect prediction will show the model's predictive accuracy. The model is most reliable for average complexity projects in the NPR 10-13 million range as seen in Figure 9, which represents the majority of residential construction in the valley. For high-cost outlier projects, the model should be used with caution and supplemented with professional judgment.

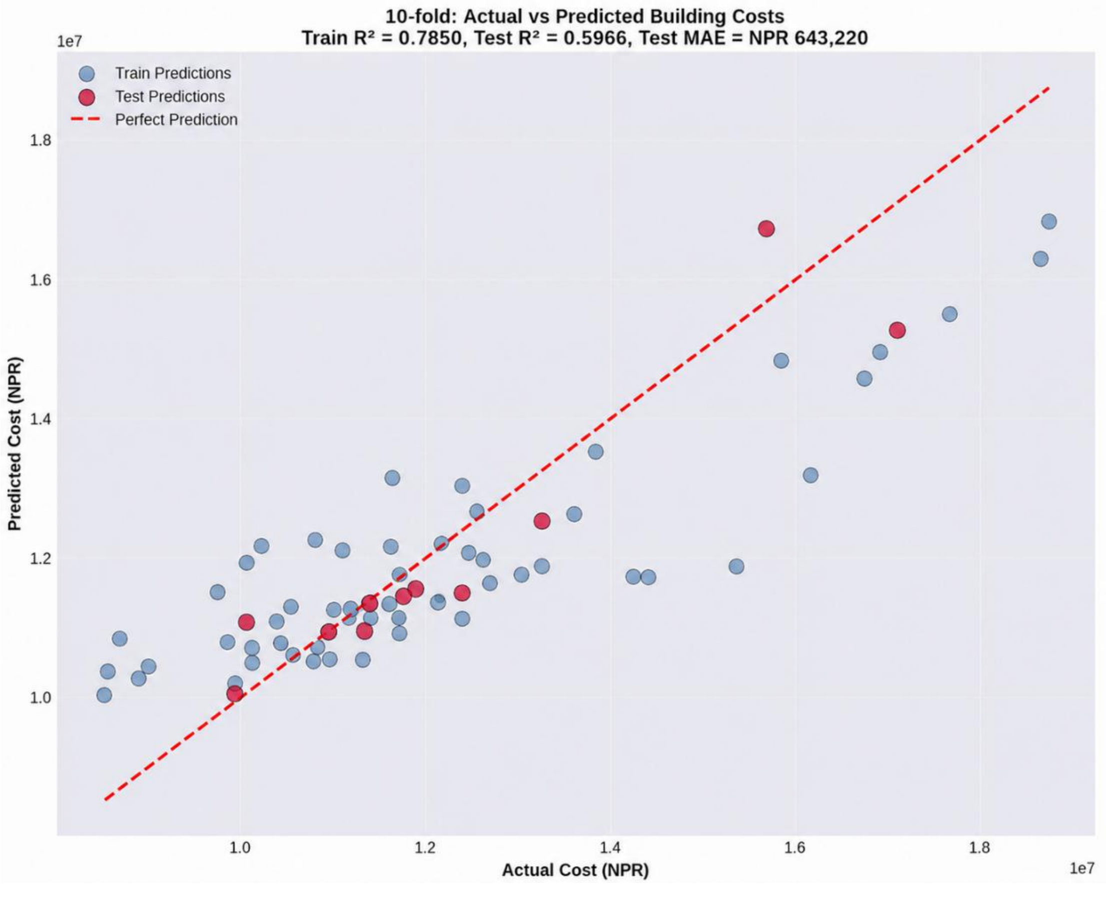

Figure 9 Actual Vs Predicted of ANN model No. 5

The ANN model thus trained can be used as a tool for prediction of buildings cost in the future. The interface shown in Figure 10 demonstrates the prediction interface, that takes as input the plinth area, number of storeys, number of columns, foundation type and location of a building, and finally outputs the estimated total cost at completion. As seen in Figure 10, the interface allows a user to input the physical dimensions of a building with numerical values, Plinth Area: 110 m^2, Number of Storeys: 2, Number of Columns: 12. The categorical features like Location: Bhaktapur, Foundation type:

Isolated, user selects from predefined categories, ensuring consistency in data formatting. Once all inputs are provided, the model encodes the data internally and processes it through the trained ANN model to generate a predicted cost. The system then displays the Estimated Total Cost of the user-defined building as 10,713,461.20. This structured and user-friendly process illustrates the practical deployment of a machine learning model for real-world cost estimation, effectively democratizing access to advanced predictive analytics for non-technical stakeholders.

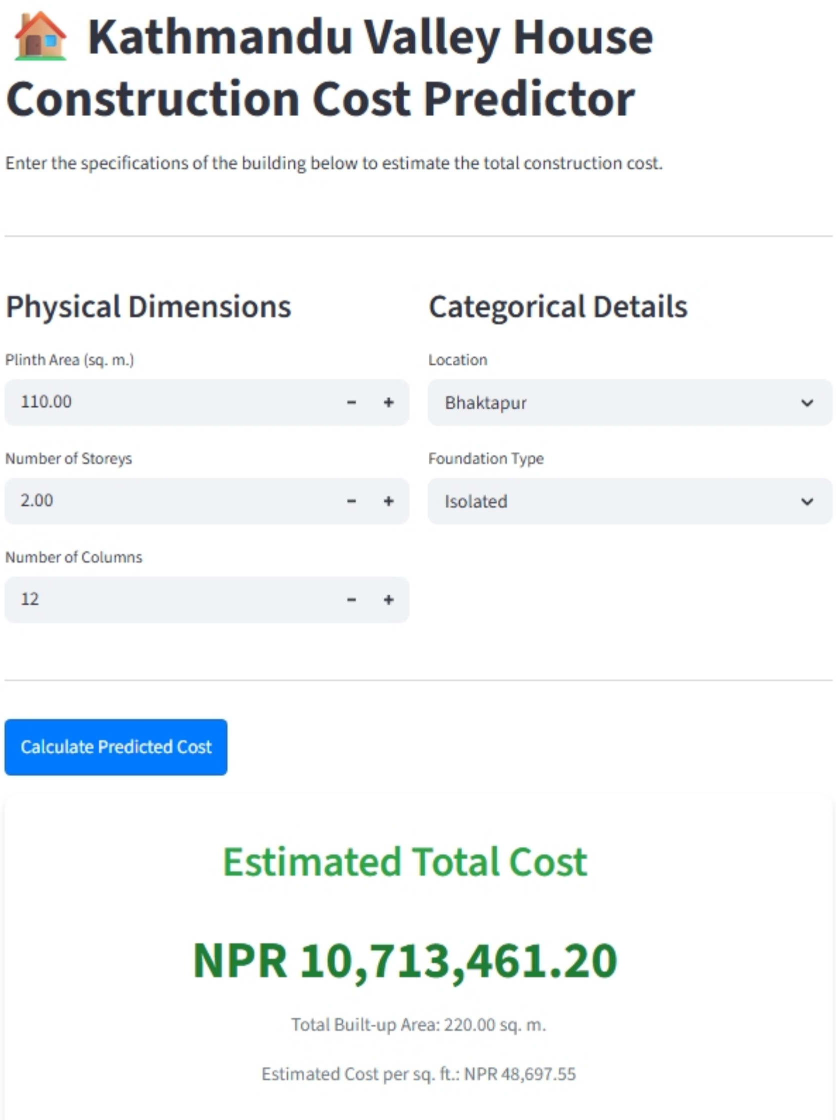

Figure 10 Cost Prediction Interface

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

This study set out to develop an Artificial Neural Network (ANN)-based model for predicting the final construction cost of residential buildings in the Kathmandu valley at an early stage of project development, using only parameters that are readily available before detailed design drawings and bill of quantities are prepared. First, the identification of input variables through a two-stage process of systematic literature review followed by structured expert consultation with professionals from DUDBC, MOUD, and contracting firms produced a validated and locally relevant set of ten input parameters: Number of Storeys, Plinth Area, Floor Height, Accessibility of Site, and Number of Columns, Classification of Building, Season of Construction Commencement, Location (district), Foundation Type, and Structural System as categorical inputs.

Statistical analysis confirmed that the number of storeys is the most dominant factor influencing cost (correlation of 0.69), followed by the plinth area. The interaction between these two variables, representing total built-up area, showed the strongest positive relationship with final construction costs.

By leveraging historical data from 76 residential projects, the study demonstrates that machine learning can significantly mitigate the inaccuracies inherent in traditional estimation methods like regression analysis or manual quantity take-offs. The optimized 10-128-64-1 architecture achieved a Mean Absolute Percentage Error (MAPE) of 7.32% on unseen testing data. With an R^2 value of 0.6787, the model proves capable of explaining nearly 68% of cost variations using only early-stage conceptual parameters. The findings validate the efficacy of Artificial Neural Networks (ANNs) as a reliable tool for preliminary cost estimation, demonstrating superior accuracy and faster turnaround compared to conventional approaches. The model is particularly reliable for average residential projects ranging between NPR 9 million and 13 million, providing stakeholders with a robust decision-support tool during the feasibility phase.

The 10-fold cross-validation process generated models with average R^2 value of 0.5211 and MAPE of 7.50%, demonstrating the robustness and generalizability of the models. While the model exhibits mild overfitting due to the relatively small dataset, it contributes a locally calibrated, data-driven estimation framework to Nepal's construction management body of knowledge and demonstrates that machine learning tools can be built and validated from the modest datasets available in Nepal's construction sector.

5.2 Recommendations

5.2.1 Recommendations from This Study

The study's key findings that serve as recommendations to practitioners are:

1. The ANN-based model (architecture: 10-128-64-1) achieved a MAPE of 7.32% and R^2 of 0.6787, making it a reliable decision-support tool for early-stage cost estimation of residential buildings in Kathmandu Valley.
2. Stakeholders (developers, consultants, government officials) should use this model during the feasibility phase for projects ranging between NPR 9 million and 13 million, where the model is most accurate.
3. The number of storeys and plinth area are the most dominant cost-influencing factors; these should be prioritized in early project scoping and budgeting decisions.
4. Machine learning tools, even when trained on modest datasets (76 samples), can outperform traditional regression and manual quantity take-off methods in preliminary cost estimation within Nepal's construction context.

5.2.2 Recommendations for Further Study

The developed model has certain limitations; therefore, the following suggestions are proposed for future researchers:

1) **Dataset Expansion and Diversification:** Future researchers should aim to collect data exceeding the 76 samples used here to improve the model's generalization and reduce overfitting. Expanding the scope beyond RCC frame structures in the Kathmandu Valley to include steel or load-bearing structures in diverse topographical regions of Nepal would create a more universally applicable tool.
2) **Inclusion of Non-Structural Variables:** As this study focused primarily on structural parameters, future models should incorporate finishing works, HVAC, electrical, and sanitary installations to provide a holistic prediction of the total project budget.
3) **Feature Enrichment:** Including additional variables such as soil type, seismic zone, labor availability, and project duration could make the model more robust, especially for detailed cost estimations at later stages.
4) **Build a dynamic, retrained tool:** The ANN model developed in this study is static i.e., trained on a fixed dataset. Future research should explore the development of a dynamic prediction framework that periodically assimilates new project completions into the training data and retrains the model at defined intervals.

`

CHAPTER 6: REFERENCES

Abdelalim, A., Salem, M., Salem, M., Al-Adwani, M., & Tantawy, M. (2025). An Analysis of Factors Contributing to Cost Overruns in the Global Construction Industry. *Buildings, 15*(1), 18. doi:https://doi.org/10.3390/buildings15010018

Acharya, B., & Karki, S. (2022, October). Artificial Neural Networks for Cost Estimation of Road Projects in. *Proceedings of 12th IOE Graduate Conference.* Kathmandu.

Afana, O., Al Zubaidi, R., Abu Dabous, S., & Ibrahim, F. (2024, December). Categories and Factors of Cost Overrun in Construction Projects: A Systematic Review. *Engineering, Technology & Applied Science Research, 14*(6), 18330-18347. doi:https://doi.org/10.48084/etasr.9006

Agarwal, R., Chandrasekaran, S., & Sridhar, M. (2016). Imagining construction's digital future. *McKinsey & Company, 24*(6), 1-13.

Al-Gahtani, K. &. (2024). ANN prediction model of final construction cost at an early stage. . *Journal of Asian Architecture and Building Engineering.* , 24. 1-25. 1.

Allan Ashworth, S. P. (2015). *Cost Studies of Buildings.* Routledge http://www.routledge.com.

Alrasheed, K. &. (2025). Artificial Neural Network-Based Cost Estimation for Public Construction Projects in Kuwait. *Journal of Engineering Research. 10.1016/j.jer.2025.09.001. .*

Arafa, M., & Alqedra, M. (2011). Early Stage Cost Estimation of Buildings Construction Projects using Artificial Neural Networks. *Journal of Artificial Intelligence 4. 10.3923/jai.2011.63.75.*

Audisio, B. (2023). Innovative Pre-Tender Costs Estimating Techniques in Construction Projects. *Ph.D. dissertation, Politecnico di Torino.*

Ayman A. Abu Hammad, S. M. (2008). Prediction Model for Construction Cost and Duration in Jordan. *Jordan Journal of Civil Engineering*, Volume 2, No. 3.

Baral, S. (2009). *Nepalese Construction Industry: Challenges.* Retrieved from santoshbaral.blogspot.com›: https://santoshbaral.blogspot.com/

Cece Suhendi, M. F. (2024). The Percentage of Structural Costs in Houses Structured Type 30, 36, 45 and 70 with Houses with Simple Structures. *Proceedings of the 2nd International Conference on Consumer Technology and Engineering Innovation (ICONTENTION 2023)* (pp. 135-143). Sukabumi, Indonesia: Atlantis Press. Retrieved from https://doi.org/10.2991/978-94-6463-406-8_24

Chao, L. C. (1998). Fuzzy Logic for Evaluating Alternative Construction Technology. *Journal of Construction Engineering and Management, 124*(4), 297-304. Retrieved from https://doi.org/10.1061/(ASCE)0733-9364(1998)124:4(297)

`

Chaulagain, H., Rodrigues, H., Spacone, E., & Varum, H. (2015). Seismic response of current RC buildings in Kathmandu Valley. *Structural Engineering and Mechanics, 53*(4), 791–818. doi:10.12989/SEM.2015.53.4.791

Cochran, W. G. (1963). *Sampling techniques. 2nd edition.* John Wiley & Sons. Retrieved from https://books.google.com.np/books?id=Y-SxXwAACAAJ

David E. Rumelhart, G. E. (1986). Learning representations by back-propagating errors. *Nature Vol. 323*, 533-536.

Dimitrijević, B., Stojadinović, Z., Marinković, D., & Dimitrijević, M. (2019). Influence of structural system on the construction time and cost of residential projects. *GRAĐEVINAR, 71 (8)*, 681-693. doi:https://doi.org/10.14256/JCE.2315.2018

Dysert, L. R. (2008). *Skills & Knowledge of Cost Engineering* (5th ed.). AACE International.

El-Sawalhi, N. I., & Shehatto, O. (2014). A Neural Network Model for Building Construction. *KICEM Journal of Construction Engineering and Project Management*. doi:http://dx.doi.org/10.6106/JCEPM.2014.4.4.009

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* Cambridge, MA, USA: The MIT Press.

Graupe, D. (2013). *Principles of artificial neural networks* (3rd ed.). World Scientific.

Hashemi, S. T., Ebadati, O. M., & Kaur, H. (2020). Cost estimation and prediction in construction projects: a systematic review on machine learning techniques. *SN Applied Sciences.*, 2. 1-27. 10.1007/. doi:https://doi.org/10.1007/s42452-020-03497-1

Hatamleh, M. T., Hiyassat, M., Sweis, G. J., & Sweis, R. J. (2018). Factors affecting the accuracy of cost estimate: case of Jordan. *Engineering, Construction and Architectural Management, 25*(1), 113-131. doi:10.1108/ECAM-10-2016-0232

Ibrahim, A. H., & Elshwadfy, L. M. (2021). Factors Affecting the Accuracy of Construction Project Cost. *Jordan Journal of Civil Engineering, 15*(3), 329-344.

Karn, S. K., & Dahal, K. R. (2021). Cost Optimization in Building Construction Projects with Special Reference to Kathmandu Valley of. *American Journal of Science, Engineering and Technology.*, Vol. 6, No. 1, 2021, pp. 8-19. doi: 10.11648/j.ajset.20210601.12.

Khalid S. Al-Gahtani, A. M. (2025). ANN prediction model of final construction cost at an early stage. *Journal of Asian Architecture and Building Engineering*,

`

24(2), 775–799. Retrieved from https://doi.org/10.1080/13467581.2023.2294883

Khanal, A. (2025, April). Early Cost Estimation of RC Bridges in Nepal using Artificial Neural Networks. *Proceedings of 16th IOE Graduate Conference.* Kathmandu.

Kim, G.-H., An, S.-H., & Kang, K.-I. (2004). Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. *Building and Environment, 39*(10), 1235-1242. doi:https://doi.org/10.1016/j.buildenv.2004.02.013.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 436–444. doi:10.1038/nature14539

Lewis, C. (1982). INDUSTRIAL AND BUSINESS FORECASTING METHODS. *Journal of Forecasting*, 144.

Liu, S. C. (2021). Application of Artificial Neural Networks in Construction Management: Current Status and Future Directions. *Applied Sciences, 11(20), 9616. https://doi.org/10.3390/app11209616*.

Magdum, S. a. (2017, 10). CONSTRUCTION COST PREDICTION USING NEURAL NETWORKS. *ICTACT Journal on Soft Computing, 8*, 1549-1556. doi:10.21917/ijsc.2017.0216

Moghayedi, A. a. (2022, 03). Modelling the uncertainty of cost and time in highway projects. *Infrastructure Asset Management*. doi:10.1680/jinam.21.00004

Sapkota, A., Karki, S., Pokharel, B., & Dhital, M. (2023). A Comparative Study of Machine Learning Algorithms for Early Cost Estimation of Building Projects in Nepal. *Kathford Journal of Engineering and Management.*, 3. 34-49. 10.31.

Shamim, M. M., Hamid, A. B., Nyamasvisva, T. E., & Rafi, N. S. (2025). Advancement of Artificial Intelligence in Cost Estimation for Project Management Success: A Systematic Review of Machine Learning, Deep Learning, Regression, and Hybrid Models. *Modelling, 6*(2), 35. doi:10.3390/modelling6020035

Theobald, O. (2021). *Machine Learning For Absolute Beginners: A Plain English Introduction.* Scatterplot press.

Veliyampatt, S. (n.d.). Determination of Efficacy of Cost Estimation Models for Building Projects using Artificial Neural Networks, Fuzzy Inference System and Regression Analysis. *International Research Journal of Engineering and Technology (IRJET), 8*(10).

Xu, H., Chang, R., Pan, M., Li, H., Liu, S., Webber, R. J., . . . Dong, a. N. (2022). Application of Artificial Neural Networks in Construction Management: A

Scientometric Review. *Buildings, 12(7), 952.*
https://doi.org/10.3390/buildings12070952.

APPENDIX

APPENDIX A: QUESTIONNAIRE FOR EXPERT OPINION

Questionnaire for Expert Opinion

Namaste! This survey is a part of Master's thesis at IOE, Pulchowk Campus, titled "ANN based prediction of final construction cost of buildings in Kathmandu Valley at an early stage". This questionnaire kindly asks your valuable time and input with the aim to identify and validate the key factors affecting the cost of building in the construction phase within Kathmandu Valley.

(Please use in Landscape Mode if you're viewing in a smartphone)

Respondent's Information

Name of the Expert: *
Name of the Company: *
Position of the Expert in the Company: *
Working experience in Building Construction Projects(years): *

Factors affecting Construction Cost

In your experience , do these factors play a role in determing the project cost of building projects in Kathmandu Valley?	Select Option	Remarks
Number of Storeys	◯ Yes ◯ No *	
Classification of Building [Hospital, Administration/Office, Halls, Quarters etc.]	◯ Yes ◯ No *	

Season of construction commencement [Summer, Autumn, Monsoon, Winter]	○ Yes ○ No *	
Plinth Area	○ Yes ○ No *	
Floor Height	○ Yes ○ No *	
Location of Building	○ Yes ○ No *	
Foundation Type [Raft, Isolated, Pile, etc.]	○ Yes ○ No *	
Accessibility of site	○ Yes ○ No *	
Structural System [R.C.C., Steel, Load Bearing, etc.]	○ Yes ○ No *	
Number of Columns	○ Yes ○ No *	

▶▶In Your opinion, are there any other parameters that determine the cost of Building projects in Kathmandu Valley? If yes, please list them out below..

`

APPENDIX B: ANALYSIS OF RESPONSE OF EXPERT OPINION

Expert	**Position**	**Work Experience in Building Project (Years)**
1	Contractor	12
2	Engineer	11
3	Engineer	7
4	Senior Divisional Engineer	20
5	Contractor	9

In your experience, do these factors play a role in determining the project cost of building projects in Kathmandu Valley?

Factors	**1**	**2**	**3**	**4**	**5**	**Count of Yes**
Number of Storeys	✓	✓	✓	✓	✓	5
Classification of Building [Hospital, Administration/Office, Halls, Quarters etc.]	✓	✓	✓	✓	✓	5
Season of construction commencement [Summer, Autumn, Monsoon, Winter]		✓		✓		2
Plinth Area	✓	✓	✓	✓	✓	5
Floor Height	✓	✓		✓	✓	4
Location of Building	✓	✓		✓	✓	4
Foundation Type [Raft, Isolated, Pile, etc.]	✓	✓	✓	✓	✓	5
Accessibility of Site	✓	✓		✓	✓	4
Structural System [R.C.C., Steel, Load Bearing, etc.]	✓	✓	✓	✓	✓	5
Number of Columns	✓	✓		✓	✓	4

`

In Your opinion, are there any other parameters that determine the cost of Building projects in Kathmandu Valley? If yes, please list them out below.

Expert 1	Expert 4	Expert 5
Quality of materials, Quality of work and Project duration	Design Factor (design considering the wastage as minimum as possible); Codal provisions; Building Byelaws; Cost of Land & Site Development works, soil bearing capacity or requirement of soil improvement measures; Labor Cost, Fuel and machinery/plants & equipment cost; Taxes, Price adjustment/escalation provision in construction material and other in contract.	Materials Quality, Design Complexity etc.

`

APPENDIX C: DISTRIBUTION OF BUILDING DATA

Figure 7-11 Distribution of Buildings Based on Construction Completion Year

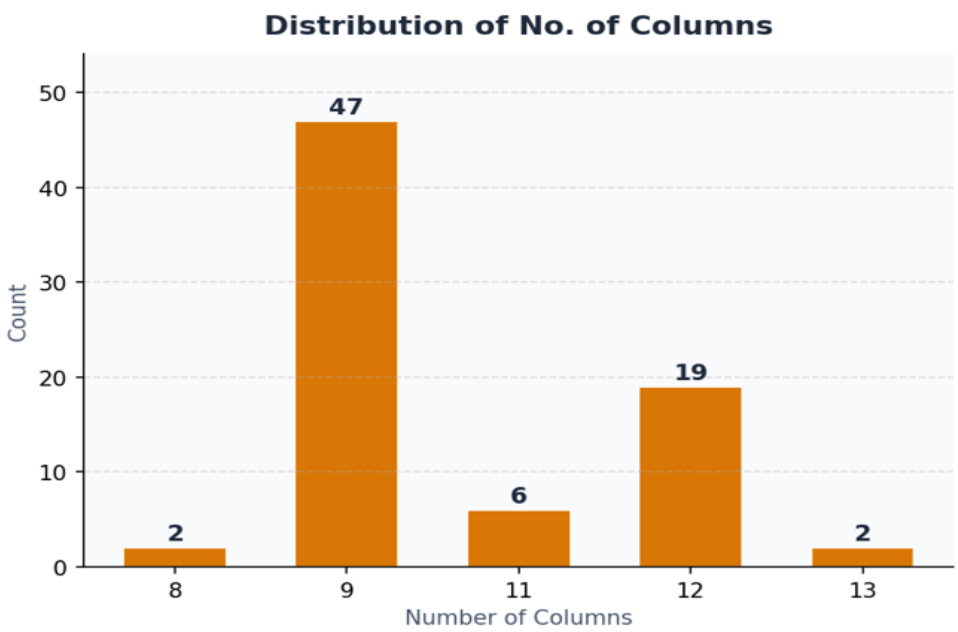

Figure 7-12 Distribution of No. of Columns

`

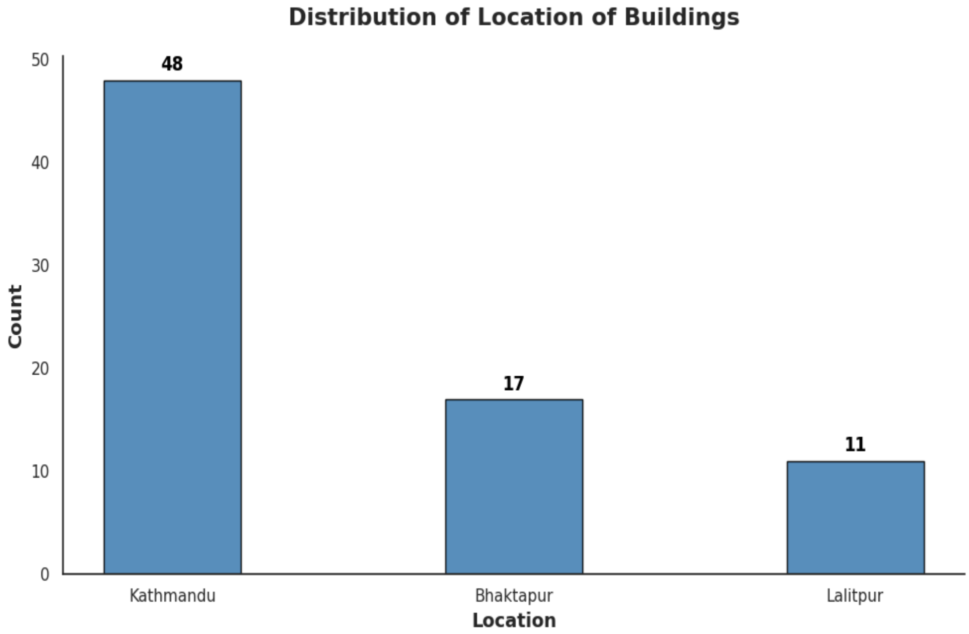

Figure 7-13 Distribution of Location of Buildings

Figure 7-14 Distribution of No. of Storey

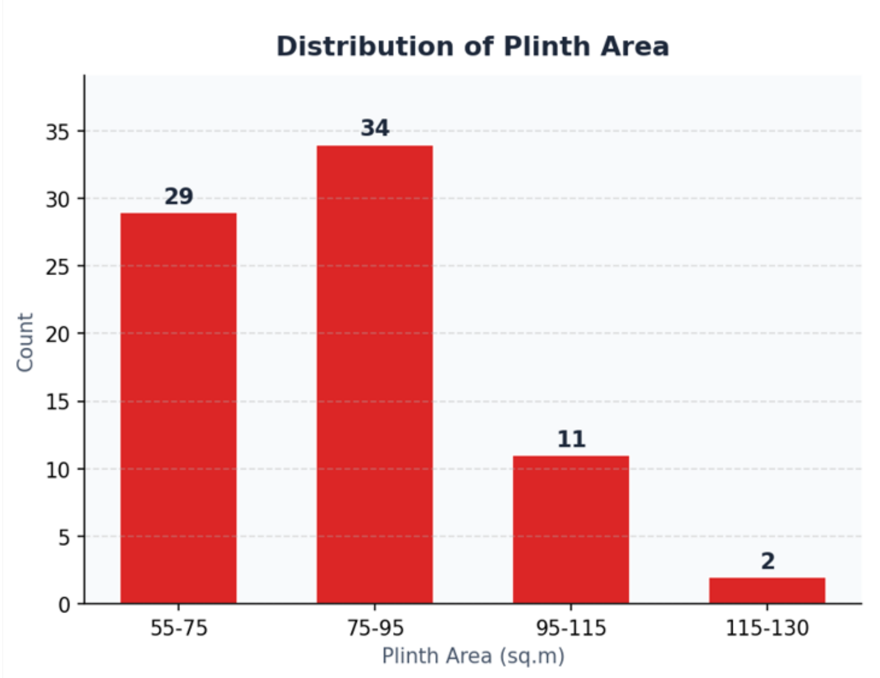

Figure 7-15 Distribution of Plinth Area

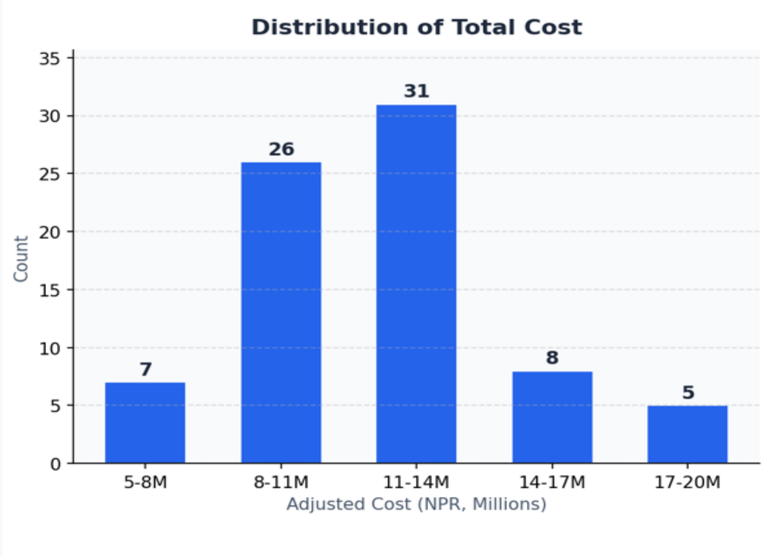

Figure 7-16 Distribution of Total Cost (NPR, Millions)

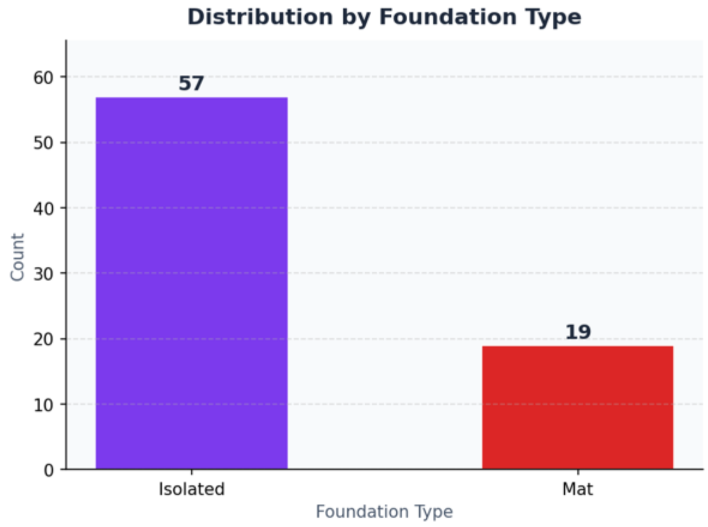

Figure 7-17 Distribution of Foundation Type

APPENDIX D: NRB WHOLESALE PRICE INDEX

67. National Wholesale Price Index (Reindexed)
Base Year : 2017/18 = 100

Fiscal Year Quarter/ Month	Overall Index	Primary	Food	Non-food	Fuel and Power	Fuel and Power	Eletricity	Manufactured	Food, Beverage and Tobacco	Textiles	Leather and Leather Products	Wood and Wood Products
	1	1.1	1.1.1	1.1.2	1.2	1.2.1	1.2.2	1.3	1.3.1	1.3.2	1.3.3	1.3.4
1999/00	**30.4**	**24.8**	**24.3**	**42.1**	**22.6**	**22.6**		**37.6**	**32.8**	**47.9**	**38.4**	
2000/01	**30.8**	**25.1**	**24.7**	**42.1**	**29.2**	**29.2**		**37.2**	**30.6**	**48.3**	**36.9**	
2001/02	**32.3**	**27.3**	**26.8**	**43.0**	**28.5**	**28.5**		**38.3**	**32.1**	**48.5**	**37.2**	
2002/03	**33.5**	**28.3**	**27.8**	**45.9**	**31.9**	**31.9**		**39.4**	**33.5**	**48.4**	**39.8**	
2003/04	**34.9**	**29.2**	**28.7**	**46.8**	**36.0**	**35.9**		**40.9**	**33.8**	**51.6**	**40.1**	
2004/05	**37.5**	**30.6**	**30.1**	**49.2**	**44.5**	**44.5**		**43.6**	**35.9**	**52.6**	**39.8**	
2005/06	**40.8**	**33.3**	**32.8**	**50.8**	**55.6**	**55.6**		**46.5**	**39.0**	**51.9**	**41.1**	
2006/07	**44.5**	**37.4**	**37.0**	**50.5**	**61.0**	**60.8**		**49.4**	**41.6**	**52.8**	**42.6**	
2007/08	**48.4**	**39.9**	**39.7**	**46.5**	**68.0**	**68.0**		**54.5**	**46.7**	**53.2**	**43.9**	
2008/09	**54.9**	**46.1**	**45.7**	**59.9**	**75.5**	**75.2**		**61.4**	**53.4**	**57.1**	**49.6**	
2009/10	**61.6**	**56.5**	**56.2**	**66.0**	**71.3**	**71.2**		**65.5**	**60.9**	**57.8**	**51.7**	
2010/11	**67.7**	**61.8**	**61.6**	**69.5**	**82.7**	**82.7**		**71.5**	**68.6**	**64.2**	**57.5**	
2011/12	**72.0**	**65.0**	**64.7**	**76.8**	**100.2**	**100.2**		**75.2**	**70.3**	**82.2**	**72.0**	
2012/13	**78.5**	**72.7**	**72.4**	**81.0**	**119.2**	**119.0**		**78.4**	**74.1**	**84.2**	**76.3**	
2013/14	**85.0**	**80.8**	**80.7**	**84.2**	**127.1**	**126.9**		**83.0**	**80.6**	**87.8**	**80.4**	
2014/15	**90.1**	**87.3**	**87.3**	**87.1**	**119.1**	**118.7**		**88.9**	**87.8**	**95.2**	**85.0**	
2015/16	**95.8**	**98.2**	**98.2**	**96.1**	**101.6**	**101.3**		**92.6**	**91.9**	**99.9**	**93.6**	
2016/17	**98.3**	**101.3**	**101.3**	**101.6**	**94.5**	**94.3**		**95.8**	**95.9**	**100.9**	**103.0**	
2017/18	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
2018/19	**106.4**	**106.4**	**106.4**	**106.3**	**112.6**	**119.5**	**100.0**	**105.3**	**105.4**	**115.1**	**104.0**	**109.4**
2019/20	**113.1**	**118.8**	**119.5**	**109.1**	**111.8**	**118.1**	**100.0**	**109.9**	**116.8**	**120.1**	**112.8**	**113.6**
2020/21	**122.1**	**130.9**	**131.4**	**123.3**	**114.5**	**122.5**	**100.0**	**117.9**	**127.0**	**127.3**	**113.1**	**117.6**
2021/22	**133.7**	**136.4**	**136.5**	**135.5**	**136.2**	**155.7**	**100.0**	**131.6**	**142.5**	**140.9**	**117.8**	**138.5**
2022/23	**145.1**	**143.8**	**143.4**	**148.2**	**168.1**	**205.4**	**100.0**	**142.3**	**154.7**	**152.8**	**128.0**	**158.2**
2023/24	**150.8**	**154.2**	**154.8**	**146.1**	**160.7**	**193.9**	**100.0**	**147.3**	**163.8**	**170.3**	**138.7**	**163.2**
I Qtr	**151.7**	**159.2**	**160.2**	**144.4**	**162.7**	**196.9**	**100.0**	**145.7**	**159.7**	**174.2**	**136.3**	**162.8**
Jul/Aug	150.1	156.6	157.5	144.7	153.6	183.0	100.0	145.8	159.9	174.2	136.3	162.8
Aug/Sep	153.1	161.7	162.9	145.0	167.3	204.2	100.0	145.9	159.6	174.2	136.3	162.8
Sep/Oct	151.9	159.2	160.4	143.4	167.5	204.4	100.0	145.3	159.6	174.2	136.3	162.8
II Qtr	**148.2**	**146.0**	**146.1**	**145.3**	**161.4**	**195.0**	**100.0**	**147.5**	**165.3**	**175.1**	**138.7**	**162.9**
Oct/Nov	150.1	150.9	151.3	145.3	163.9	198.9	100.0	147.6	165.7	175.1	138.7	162.9
Nov/Dec	147.6	144.3	144.2	145.3	160.6	193.8	100.0	147.6	165.3	175.1	138.7	162.9
Dec/Jan	147.0	143.0	142.8	145.3	159.7	192.4	100.0	147.4	165.0	175.1	138.7	162.9
III Qtr	**149.6**	**151.4**	**151.7**	**146.8**	**160.7**	**193.9**	**100.0**	**146.8**	**164.7**	**166.3**	**138.9**	**162.8**
Jan/Feb	147.5	146.1	146.1	146.6	160.3	193.4	100.0	146.3	164.4	166.3	138.9	162.8
Feb/Mar	149.2	150.4	150.6	146.6	161.8	195.6	100.0	146.6	164.9	166.3	138.9	162.8
Mar/Apr	152.1	158.0	158.8	147.2	159.8	192.6	100.0	147.5	164.9	166.3	138.9	162.8
IV Qtr	**153.6**	**160.1**	**161.0**	**147.8**	**157.9**	**189.7**	**100.0**	**149.2**	**165.4**	**165.9**	**141.0**	**164.2**
Apr/May	153.9	161.3	162.3	148.1	160.9	194.3	100.0	148.5	165.0	165.9	141.0	164.2
May/Jun	152.6	156.8	157.5	148.1	156.4	187.3	100.0	149.6	165.4	165.9	141.0	164.2
Jun/Jul	154.4	162.2	163.3	147.3	156.6	187.6	100.0	149.6	165.7	165.9	141.0	164.2
2024/25	**156.6**	**165.4**	**166.9**	**145.8**	**154.4**	**184.2**	**100.0**	**151.8**	**169.4**	**168.5**	**142.5**	**165.4**
I Qtr	**157.7**	**172.6**	**174.5**	**146.6**	**155.9**	**186.6**	**100.0**	**149.4**	**166.9**	**166.6**	**141.5**	**164.1**
Jul/Aug	155.6	166.4	167.9	146.5	158.7	190.8	100.0	148.8	166.6	166.6	141.5	164.1
Aug/Sep	157.3	171.4	173.3	146.4	155.2	185.5	100.0	149.4	166.8	166.6	141.5	164.1
Sep/Oct	160.3	179.8	182.3	146.9	153.9	183.4	100.0	149.8	167.5	166.6	141.5	164.1
II Qtr	**156.0**	**166.9**	**168.4**	**145.6**	**154.8**	**184.8**	**100.0**	**149.9**	**167.5**	**168.3**	**142.3**	**165.0**
Oct/Nov	157.9	172.0	173.9	146.3	154.0	183.7	100.0	150.2	166.4	168.3	142.3	165.0
Nov/Dec	157.3	170.8	172.7	145.3	155.4	185.7	100.0	149.6	167.4	168.3	142.3	165.0
Dec/Jan	152.9	157.7	158.7	145.2	155.0	185.1	100.0	149.7	168.7	168.3	142.3	165.0
III Qtr	**155.6**	**159.8**	**160.9**	**145.4**	**156.1**	**186.9**	**100.0**	**153.1**	**171.2**	**169.4**	**143.0**	**165.8**
Jan/Feb	152.6	152.5	153.1	145.3	157.3	188.6	100.0	151.9	170.5	169.4	143.0	165.8
Feb/Mar	155.8	159.6	160.7	145.3	157.4	188.8	100.0	153.3	171.0	169.4	143.0	165.8
Mar/Apr	158.5	167.2	168.8	145.7	153.7	183.2	100.0	154.1	172.1	169.4	143.0	165.8
IV Qtr	**157.0**	**162.6**	**163.8**	**145.7**	**150.7**	**178.5**	**100.0**	**154.7**	**172.0**	**169.6**	**143.3**	**166.5**
Apr/May	159.9	170.8	172.8	144.2	151.3	179.5	100.0	154.9	172.1	169.6	143.3	166.5
May/Jun	155.0	157.4	158.2	146.1	150.0	177.4	100.0	154.4	172.3	169.6	143.3	166.5
Jun/Jul	156.1	159.5	160.5	146.8	150.9	178.8	100.0	154.8	171.5	169.6	143.3	166.5
2025/26												
I Qtr	**160.8**	**167.5**	**168.5**	**154.5**	**151.8**	**180.2**	**100.0**	**158.2**	**174.7**	**176.4**	**144.8**	**169.1**
Jul/Aug	159.3	165.5	166.4	153.8	153.7	183.2	100.0	156.5	174.1	176.4	144.8	169.1
Aug/Sep	160.6	167.9	168.9	153.8	150.9	178.8	100.0	157.9	174.7	176.4	144.8	169.1
Sep/Oct	162.4	169.2	170.2	155.7	150.9	178.8	100.0	160.1	175.2	176.4	144.8	169.1

`

APPENDIX E: PYTHON CODE

SECTION 1: IMPORTS AND CONFIGURATION

```
import sys
import os
import warnings

 Set UTF-8 encoding for output
sys.stdout.reconfigure(encoding='utf-8')
sys.stderr.reconfigure(encoding='utf-8')

 Redirect output to file
output_file = "85_80_70_combined_output.txt"
original_stdout = sys.stdout
sys.stdout = open(output_file, 'w', encoding='utf-8')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler,
PolynomialFeatures, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import (
    mean_absolute_error, mean_squared_error, r2_score,
    mean_absolute_percentage_error,
explained_variance_score
)
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, regularizers,
callbacks

 Suppress warnings for cleaner output
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
warnings.filterwarnings("ignore")

 Set random seeds for reproducibility
np.random.seed(42)
```

`

```
tf.random.set_seed(42)

 Set style for professional plots
plt.style.use('seaborn-v0_8-darkgrid')
sns.set_palette("husl")
plt.rcParams['figure.dpi'] = 300
plt.rcParams['savefig.dpi'] = 300
plt.rcParams['font.size'] = 10
plt.rcParams['axes.titlesize'] = 12
plt.rcParams['axes.labelsize'] = 11
plt.rcParams['legend.fontsize'] = 10

 Create output directory
output_dir = "thesis_figures"
os.makedirs(output_dir, exist_ok=True)

print("="*80)
print(" " * 15 + "THESIS RESEARCH: BUILDING COST
PREDICTION MODEL")
print("="*80)
print("Artificial Neural Network (ANN) Model Analysis")
print("="*80)
print("\n*** COMBINED ANN MODELS: 85:15, 80:20, and 70:30
splits ***")
print("="*80)


 SECTION 2: DATA LOADING AND PREPROCESSING


print("\n" + "="*80)
print("SECTION 2: DATA LOADING AND PREPROCESSING")
print("="*80)

 Load dataset
df_raw = pd.read_csv("Building_Data_v2.csv")
TARGET = "Adjusted Cost (Base Year 2025/26)"
print(f"\n[OK] Dataset loaded successfully")
print(f"  - Original samples: {len(df_raw):,}")
print(f"  - Features: {len(df_raw.columns)}")

 Clean target variable
df_raw[TARGET] = pd.to_numeric(
    df_raw[TARGET].astype(str).str.replace(",",
```

`

```
"").str.strip(),
    errors="coerce"
)
df_raw = df_raw.dropna(subset=[TARGET])
print(f"  - After cleaning: {len(df_raw):,}")

 Outlier removal (consistent for all models)
q_low = df_raw[TARGET].quantile(0.10)
q_hi = df_raw[TARGET].quantile(0.99)
outliers_removed = len(df_raw) -
len(df_raw[(df_raw[TARGET] < q_hi) & (df_raw[TARGET] >
q_low)])
df = df_raw[(df_raw[TARGET] < q_hi) & (df_raw[TARGET] >
q_low)].copy()
print(f"  - Outliers removed (bottom 10%, top 1%):
{outliers_removed}
({outliers_removed/len(df_raw)*100:.1f}%)")
print(f"  - Final dataset size: {len(df):,}")

 Feature engineering
df["Total_Area"] = df["Plinth Area"] * df["No. of
Storeys"]
df["Col_Intensity"] = df["No. of Columns"] / df["Plinth
Area"]


 SECTION 3: EXPLORATORY DATA ANALYSIS (EDA)


print("\n" + "="*80)
print("SECTION 3: EXPLORATORY DATA ANALYSIS")
print("="*80)

 Table 1: Descriptive Statistics
desc_stats = df[TARGET].describe()
table1_desc_stats = pd.DataFrame({
    'Statistic': ['Count', 'Mean (NPR)', 'Std Dev (NPR)',
'Min (NPR)', '25th Percentile',
                  'Median (NPR)', '75th Percentile', 'Max
(NPR)', 'Skewness', 'Kurtosis'],
    'Value': [
        f'{desc_stats["count"]:,.0f}',
        f'{desc_stats["mean"]:,.0f}',
        f'{desc_stats["std"]:,.0f}',
```

`

```
        f'{desc_stats["min"]:,.0f}',
        f'{desc_stats["25%"]:,.0f}',
        f'{desc_stats["50%"]:,.0f}',
        f'{desc_stats["75%"]:,.0f}',
        f'{desc_stats["max"]:,.0f}',
        f'{df[TARGET].skew():.3f}',
        f'{df[TARGET].kurtosis():.3f}'
    ]
})
print("\n[TABLE] TABLE 1: Descriptive Statistics of
Building Costs")
print(table1_desc_stats.to_string(index=False))
table1_desc_stats.to_csv(f"{output_dir}/table1_desc_stats
.csv", index=False)

 Table 2: Categorical Variable Distribution
location_dist = df['Location'].value_counts()
foundation_dist = df['Foundation Type'].value_counts()
print("\n[TABLE] TABLE 2: Categorical Variable
Distribution")
print("\nLocation Distribution:")
print(location_dist.to_string())
print("\nFoundation Type Distribution:")
print(foundation_dist.to_string())

 Figure 1: Cost Distribution Histogram
fig1, axes = plt.subplots(1, 2, figsize=(14, 5))
axes[0].hist(df[TARGET], bins=25, edgecolor='black',
alpha=0.7, color='steelblue')
axes[0].axvline(df[TARGET].mean(), color='red',
linestyle='--', linewidth=2, label=f'Mean:
{df[TARGET].mean():,.0f}')
axes[0].axvline(df[TARGET].median(), color='green',
linestyle='--', linewidth=2, label=f'Median:
{df[TARGET].median():,.0f}')
axes[0].set_xlabel('Building Cost (NPR)',
fontweight='bold')
axes[0].set_ylabel('Frequency', fontweight='bold')
axes[0].set_title('Distribution of Building Costs',
fontweight='bold')
axes[0].legend()
axes[0].grid(True, alpha=0.3)

 Box plot by location
```

`

```
locations = df['Location'].unique()
location_data = [df[df['Location'] == loc][TARGET] for
loc in locations]
bp = axes[1].boxplot(location_data, labels=locations,
patch_artist=True)
for patch in bp['boxes']:
    patch.set_facecolor('lightblue')
axes[1].set_xlabel('Location', fontweight='bold')
axes[1].set_ylabel('Building Cost (NPR)',
fontweight='bold')
axes[1].set_title('Cost Distribution by Location',
fontweight='bold')
axes[1].tick_params(axis='x', rotation=45)
axes[1].grid(True, alpha=0.3)

plt.suptitle('Figure 1: Building Cost Analysis',
fontsize=14, fontweight='bold')
plt.tight_layout()
plt.savefig(f"{output_dir}/figure1_cost_distribution.png"
, dpi=300, bbox_inches='tight')
plt.close()
print("\n[OK] Figure 1 saved:
figure1_cost_distribution.png")


 SECTION 4: ARTIFICIAL NEURAL NETWORK (ANN) MODELS - ALL
THREE SPLITS


print("\n" + "="*80)
print("SECTION 4: ARTIFICIAL NEURAL NETWORK (ANN) MODEL
DEVELOPMENT")
print("="*80)
print("*** Training THREE ANN models with 85:15, 80:20,
and 70:30 splits ***")

 Prepare data for ANN with polynomial features
ann_features = ["Total_Area", "No. of Storeys", "Plinth
Area", "No. of Columns", "Foundation Type", "Location"]
X_ann = df[ann_features]
y_ann = df[TARGET].values.reshape(-1, 1)

 Scale target variable
scaler_y = StandardScaler()
```

`

```
y_ann_scaled = scaler_y.fit_transform(y_ann)

 Create preprocessor for ANN
numeric_transformer = Pipeline([
    ("poly", PolynomialFeatures(degree=2,
include_bias=False)),
    ("scaler", StandardScaler())
])

preprocessor = ColumnTransformer([
    ("num", numeric_transformer, ["Total_Area", "No. of
Storeys", "Plinth Area", "No. of Columns"]),
    ("cat", OneHotEncoder(handle_unknown="ignore",
sparse_output=False), ["Foundation Type", "Location"])
])

 Define ALL THREE train-test splits (including 85:15)
splits = [
    (0.15, "85:15"),
    (0.20, "80:20"),
    (0.30, "70:30")
]

ann_results = {}

 Store pre-trained 85:15 results (from previous training)
 MAPE: 4.97% (train), 7.32% (test)
pretrained_85_15 = {
    'train_mape': 4.97,
    'test_mape': 7.32
}

for test_size, split_name in splits:
    print(f"\n{'='*60}")
    print(f"TRAINING ANN MODEL WITH {split_name} SPLIT")
    print(f"{'='*60}")

     Split data
    X_ann_train, X_ann_test, y_ann_train, y_ann_test =
train_test_split(
        X_ann, y_ann_scaled, test_size=test_size,
random_state=42
    )
```

`

```
    print(f"\n[DATA] Split Configuration:")
    print(f"  - Training samples: {len(X_ann_train)}
({(1-test_size)*100:.0f}%)")
    print(f"  - Testing samples: {len(X_ann_test)}
({test_size*100:.0f}%)")

     Transform features
    X_ann_train_scaled =
preprocessor.fit_transform(X_ann_train)
    X_ann_test_scaled =
preprocessor.transform(X_ann_test)

     ANN Model Architecture
    if split_name == "85:15":   Print architecture only
once
        print("\n[TOOLS] ANN Model Architecture
Specifications:")
        print(f"  - Input layer:
{X_ann_train_scaled.shape[1]} neurons")
        print(f"  - Hidden Layer 1: 128 neurons with ReLU
activation + L2 regularization (0.005)")
        print(f"  - Dropout Layer 1: 20% dropout rate")
        print(f"  - Hidden Layer 2: 64 neurons with ReLU
activation + L2 regularization (0.005)")
        print(f"  - Output Layer: 1 neuron (linear
activation)")
        print(f"  - Total trainable parameters: {128 *
X_ann_train_scaled.shape[1] + 128 + 128 * 64 + 64 + 64 *
1 + 1:,}")
        print(f"  - Optimizer: Adam
(learning_rate=0.001)")
        print(f"  - Loss function: Mean Squared Error
(MSE)")
        print(f"  - Regularization: L2 (0.005) on hidden
layers")
        print(f"  - Dropout rate: 0.2")
        print(f"  - Batch size: 8")
        print(f"  - Maximum epochs: 500")
        print(f"  - Early stopping patience: 50")
        print(f"  - Learning rate reduction patience:
15")

     Build ANN Model
    ann_model = keras.Sequential([
```

`

```
layers.Input(shape=(X_ann_train_scaled.shape[1],)),
        layers.Dense(128, activation='relu',
kernel_regularizer=regularizers.l2(0.005)),
        layers.Dropout(0.2),
        layers.Dense(64, activation='relu',
kernel_regularizer=regularizers.l2(0.005)),
        layers.Dense(1)
    ])

    ann_model.compile(

optimizer=keras.optimizers.Adam(learning_rate=0.001),
        loss='mse',
        metrics=['mae']
    )

     Callbacks
    lr_callback = callbacks.ReduceLROnPlateau(
        monitor='val_loss', factor=0.5, patience=15,
min_lr=0.00001, verbose=0
    )
    early_stop = callbacks.EarlyStopping(
        monitor='val_loss', patience=50,
restore_best_weights=True, verbose=0
    )

    print(f"\n[RUN] Training ANN Model
({split_name})...")
    ann_history = ann_model.fit(
        X_ann_train_scaled, y_ann_train,
        validation_data=(X_ann_test_scaled, y_ann_test),
        epochs=500,
        batch_size=8,
        verbose=0,
        callbacks=[lr_callback, early_stop]
    )

     Save the model
    model_filename = f"ann_model_{split_name.replace(':',
'_')}.h5"
    ann_model.save(f"{output_dir}/{model_filename}")
    print(f"  - Model saved as: {model_filename}")
```

`

```
     ANN Predictions
    ann_pred_train_scaled =
ann_model.predict(X_ann_train_scaled, verbose=0)
    ann_pred_test_scaled =
ann_model.predict(X_ann_test_scaled, verbose=0)

     Inverse transform predictions
    ann_pred_train =
scaler_y.inverse_transform(ann_pred_train_scaled).flatten
()
    ann_pred_test =
scaler_y.inverse_transform(ann_pred_test_scaled).flatten(
)
    y_ann_train_actual =
scaler_y.inverse_transform(y_ann_train).flatten()
    y_ann_test_actual =
scaler_y.inverse_transform(y_ann_test).flatten()

     ANN Performance Metrics - Training Data
    ann_train_r2 = r2_score(y_ann_train_actual,
ann_pred_train)
    ann_train_mape =
mean_absolute_percentage_error(y_ann_train_actual,
ann_pred_train) * 100
    ann_train_mae =
mean_absolute_error(y_ann_train_actual, ann_pred_train)
    ann_train_rmse =
np.sqrt(mean_squared_error(y_ann_train_actual,
ann_pred_train))

     ANN Performance Metrics - Testing Data
    ann_test_r2 = r2_score(y_ann_test_actual,
ann_pred_test)
    ann_test_mape =
mean_absolute_percentage_error(y_ann_test_actual,
ann_pred_test) * 100
    ann_test_mae = mean_absolute_error(y_ann_test_actual,
ann_pred_test)
    ann_test_rmse =
np.sqrt(mean_squared_error(y_ann_test_actual,
ann_pred_test))

    ann_metrics = {
        'R² Score': ann_test_r2,
```

`

```
        'Adjusted R²': 1 - (1 - ann_test_r2) *
(len(y_ann_test_actual) - 1) / (len(y_ann_test_actual) -
X_ann_train_scaled.shape[1] - 1),
        'MAE (NPR)': ann_test_mae,
        'RMSE (NPR)': ann_test_rmse,
        'MAPE (%)': ann_test_mape,
        'Explained Variance':
explained_variance_score(y_ann_test_actual,
ann_pred_test)
    }

     Store results
    ann_results[split_name] = {
        'metrics': ann_metrics,
        'train_metrics': {
            'R² Score': ann_train_r2,
            'MAE (NPR)': ann_train_mae,
            'RMSE (NPR)': ann_train_rmse,
            'MAPE (%)': ann_train_mape
        },
        'history': ann_history,
        'samples': {
            'train': len(X_ann_train),
            'test': len(X_ann_test)
        },
        'predictions': {
            'train': ann_pred_train,
            'test': ann_pred_test,
            'actual_train': y_ann_train_actual,
            'actual_test': y_ann_test_actual
        }
    }

    print(f"\n[TABLE] ANN Performance Metrics
({split_name}) - TRAINING Data:")
    print(f"  - R² Score: {ann_train_r2:.4f}")
    print(f"  - MAE: NPR {ann_train_mae:,.0f}")
    print(f"  - RMSE: NPR {ann_train_rmse:,.0f}")
    print(f"  - MAPE: {ann_train_mape:.2f}%")

    print(f"\n[TABLE] ANN Performance Metrics
({split_name}) - TESTING Data:")
    print(f"  - R² Score: {ann_test_r2:.4f}")
    print(f"  - Adjusted R²: {ann_metrics['Adjusted
```

`

```
R²']:.4f}")
    print(f"  - MAE: NPR {ann_test_mae:,.0f}")
    print(f"  - RMSE: NPR {ann_test_rmse:,.0f}")
    print(f"  - MAPE: {ann_test_mape:.2f}%")
    print(f"  - Explained Variance:
{ann_metrics['Explained Variance']:.4f}")

     Training history
    print(f"\n[STATS] ANN Training Summary
({split_name}):")
    print(f"  - Final training loss:
{ann_history.history['loss'][-1]:.4f}")
    print(f"  - Final validation loss:
{ann_history.history['val_loss'][-1]:.4f}")
    print(f"  - Best validation loss:
{min(ann_history.history['val_loss']):.4f}")
    print(f"  - Epochs trained:
{len(ann_history.history['loss'])}")
    print(f"  - Early stopping triggered: {'Yes' if
len(ann_history.history['loss']) < 500 else 'No'}
(patience=50)")


 SECTION 5: PRE-TRAINED 85:15 RESULTS DOCUMENTATION


print("\n" + "="*80)
print("SECTION 5: PRE-TRAINED ANN (85:15) RESULTS")
print("="*80)
print("*** Note: These results are from previous training
run ***")
print(f"  - Training MAPE:
{pretrained_85_15['train_mape']:.2f}%")
print(f"  - Testing MAPE:
{pretrained_85_15['test_mape']:.2f}%")

 Add pre-trained results to ann_results for documentation
print("\n[TABLE] ANN (85:15) - PRE-TRAINED RESULTS (from
previous run):")
print(f"  - Training MAPE:
{pretrained_85_15['train_mape']:.2f}%")
print(f"  - Testing MAPE:
{pretrained_85_15['test_mape']:.2f}%")
print("  - Note: R² and other metrics were computed in
```

`

```
the original training run")


 SECTION 6: MODEL COMPARISON TABLE - ALL ANN SPLITS


print("\n" + "="*80)
print("SECTION 6: MODEL PERFORMANCE COMPARISON - ALL
THREE ANN SPLITS")
print("="*80)

comparison_table = pd.DataFrame({
    'Metric': ['R² Score', 'Adjusted R²', 'MAE (NPR)',
'RMSE (NPR)', 'MAPE (%)', 'Explained Variance'],
    'ANN (85:15)*': [
        'N/A (pre-trained)',
        'N/A (pre-trained)',
        'N/A (pre-trained)',
        'N/A (pre-trained)',
        f'{pretrained_85_15["test_mape"]:.2f}%',
        'N/A (pre-trained)'
    ],
    'ANN (80:20)': [
        f'{ann_results["80:20"]["metrics"]["R²
Score"]:.4f}',
        f'{ann_results["80:20"]["metrics"]["Adjusted
R²"]:.4f}',
        f'{ann_results["80:20"]["metrics"]["MAE
(NPR)"]:,.0f}',
        f'{ann_results["80:20"]["metrics"]["RMSE
(NPR)"]:,.0f}',
        f'{ann_results["80:20"]["metrics"]["MAPE
(%)"]:.2f}%',
        f'{ann_results["80:20"]["metrics"]["Explained
Variance"]:.4f}'
    ],
    'ANN (70:30)': [
        f'{ann_results["70:30"]["metrics"]["R²
Score"]:.4f}',
        f'{ann_results["70:30"]["metrics"]["Adjusted
R²"]:.4f}',
        f'{ann_results["70:30"]["metrics"]["MAE
(NPR)"]:,.0f}',
        f'{ann_results["70:30"]["metrics"]["RMSE
```

`

```
(NPR)"]:,.0f}',
        f'{ann_results["70:30"]["metrics"]["MAPE
(%)"]:.2f}%',
        f'{ann_results["70:30"]["metrics"]["Explained
Variance"]:.4f}'
    ]
})

print("\n[TABLE] TABLE 3: Comparative Model Performance
(All ANN Splits)")
print(comparison_table.to_string(index=False))
comparison_table.to_csv(f"{output_dir}/table3_model_compa
rison.csv", index=False)


 SECTION 7: ANN SPLIT COMPARISON SUMMARY


print("\n" + "="*80)
print("SECTION 7: ANN TRAIN-TEST SPLIT COMPARISON")
print("="*80)

 Create detailed comparison for ANN splits
ann_split_comparison = pd.DataFrame({
    'Split': ['85:15', '80:20', '70:30'],
    'Training %': ['85%', '80%', '70%'],
    'Testing %': ['15%', '20%', '30%'],
    'Training Samples': [
        f'{int(len(X_ann) * 0.85):,}',
        f'{int(len(X_ann) * 0.80):,}',
        f'{int(len(X_ann) * 0.70):,}'
    ],
    'Testing Samples': [
        f'{int(len(X_ann) * 0.15):,}',
        f'{int(len(X_ann) * 0.20):,}',
        f'{int(len(X_ann) * 0.30):,}'
    ],
    'Training MAPE (%)': [
        f'{pretrained_85_15["train_mape"]:.2f}',
        f'{ann_results["80:20"]["train_metrics"]["MAPE
(%)"]:.2f}',
        f'{ann_results["70:30"]["train_metrics"]["MAPE
(%)"]:.2f}'
    ],
```

`

```
    'Testing MAPE (%)': [
        f'{pretrained_85_15["test_mape"]:.2f}',
        f'{ann_results["80:20"]["metrics"]["MAPE
(%)"]:.2f}',
        f'{ann_results["70:30"]["metrics"]["MAPE
(%)"]:.2f}'
    ],
    'Testing R²': [
        'N/A (pre-trained)',
        f'{ann_results["80:20"]["metrics"]["R²
Score"]:.4f}',
        f'{ann_results["70:30"]["metrics"]["R²
Score"]:.4f}'
    ]
})

print("\n[TABLE] TABLE 4: ANN Train-Test Split
Comparison")
print(ann_split_comparison.to_string(index=False))
ann_split_comparison.to_csv(f"{output_dir}/table4_ann_spl
it_comparison.csv", index=False)

 SECTION 8: 10-Fold Cross-Validation


kf = KFold(n_splits=10, shuffle=True, random_state=42)


fold_metrics = {'MAPE': [], 'MSE': [], 'RMSE': [], 'R2':
[]}

all_y_true, all_y_pred, histories = [], [], []


print("\nStarting 10-Fold Cross Validation...")


 We fit the preprocessor on the whole dataset once just
to get the input_dim and feature names

X_transformed_full = preprocessor.fit_transform(X)

input_dim = X_transformed_full.shape[1]
```

`

```
feature_names = numerical_cols +
list(preprocessor.named_transformers_['cat'].get_feature_
names_out(categorical_cols))


fold = 1
for train_index, val_index in kf.split(X):
     Split Data
    X_train, X_val = X.iloc[train_index],
X.iloc[val_index]
    y_train, y_val = y.iloc[train_index],
y.iloc[val_index]


     Preprocess features (fit on train, transform on
validation)
    X_train_processed =
preprocessor.fit_transform(X_train)
    X_val_processed = preprocessor.transform(X_val)


     Build Model
    model = build_ann_model(input_dim)


     Train Model
    history = model.fit(
        X_train_processed, y_train,
        epochs=150,
        batch_size=8,
        verbose=0,  Set to 1 to see epoch progress
        validation_data=(X_val_processed, y_val)
    )
    histories.append(history)
```

`

```
     Predictions
    y_pred = model.predict(X_val_processed,
verbose=0).flatten()

     Calculate Metrics
    mape = mean_absolute_percentage_error(y_val, y_pred)
    mse = mean_squared_error(y_val, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_val, y_pred)

     Store metrics
    fold_metrics['MAPE'].append(mape)
    fold_metrics['MSE'].append(mse)
    fold_metrics['RMSE'].append(rmse)
    fold_metrics['R2'].append(r2)

    all_y_true.extend(y_val)
    all_y_pred.extend(y_pred)

    print(f"Fold {fold}: MAPE = {mape:.4f} | RMSE =
{rmse:,.0f} | R2 = {r2:.4f}")
    fold += 1


 SECTION 9: FIGURES

print("\n" + "="*80)
print("SECTION 8: GENERATING THESIS FIGURES")
print("="*80)

 Figure 2: Actual vs Predicted - ANN (80:20)
fig2, ax = plt.subplots(figsize=(10, 8))
ax.scatter(ann_results["80:20"]["predictions"]["actual_te
st"],
```

`

```
            ann_results["80:20"]["predictions"]["test"],
            alpha=0.6, edgecolors='black', linewidth=0.5,
s=100, color='green', label='Predictions')
ax.plot([ann_results["80:20"]["predictions"]["actual_test
"].min(),

ann_results["80:20"]["predictions"]["actual_test"].max()]
,

[ann_results["80:20"]["predictions"]["actual_test"].min()
,

ann_results["80:20"]["predictions"]["actual_test"].max()]
,
        'r--', lw=2, label='Perfect Prediction')
ax.set_xlabel('Actual Cost (NPR)', fontweight='bold')
ax.set_ylabel('Predicted Cost (NPR)', fontweight='bold')
ax.set_title(f'ANN (80:20): Actual vs Predicted Building
Costs\nR² = {ann_results["80:20"]["metrics"]["R²
Score"]:.4f}, MAPE =
{ann_results["80:20"]["metrics"]["MAPE (%)"]:.2f}%',
fontweight='bold')
ax.legend()
ax.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig(f"{output_dir}/figure2_ann_actual_vs_predicte
d.png", dpi=300, bbox_inches='tight')
plt.close()
print("[OK] Figure 2 saved:
figure2_ann_actual_vs_predicted.png")

 Figure 3: Actual vs Predicted - ANN (70:30)
fig3, ax = plt.subplots(figsize=(10, 8))
ax.scatter(ann_results["70:30"]["predictions"]["actual_te
st"],
            ann_results["70:30"]["predictions"]["test"],
            alpha=0.6, edgecolors='black', linewidth=0.5,
s=100, color='purple', label='Predictions')
ax.plot([ann_results["70:30"]["predictions"]["actual_test
"].min(),

ann_results["70:30"]["predictions"]["actual_test"].max()]
,
```

`

```
[ann_results["70:30"]["predictions"]["actual_test"].min()
,

ann_results["70:30"]["predictions"]["actual_test"].max()]
,
        'r--', lw=2, label='Perfect Prediction')
ax.set_xlabel('Actual Cost (NPR)', fontweight='bold')
ax.set_ylabel('Predicted Cost (NPR)', fontweight='bold')
ax.set_title(f'ANN (70:30): Actual vs Predicted Building
Costs\nR² = {ann_results["70:30"]["metrics"]["R²
Score"]:.4f}, MAPE =
{ann_results["70:30"]["metrics"]["MAPE (%)"]:.2f}%',
fontweight='bold')
ax.legend()
ax.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig(f"{output_dir}/figure3_ann_70_30_actual_vs_pr
edicted.png", dpi=300, bbox_inches='tight')
plt.close()
print("[OK] Figure 3 saved:
figure3_ann_70_30_actual_vs_predicted.png")

 Figure 4: ANN Split Comparison - MAPE
fig4, axes = plt.subplots(1, 2, figsize=(14, 5))

splits_labels = ['85:15\n(Pre-trained)', '80:20',
'70:30']
train_mapes = [pretrained_85_15['train_mape'],
ann_results["80:20"]["train_metrics"]["MAPE (%)"],

ann_results["70:30"]["train_metrics"]["MAPE (%)"]]
test_mapes = [pretrained_85_15['test_mape'],
ann_results["80:20"]["metrics"]["MAPE (%)"],
              ann_results["70:30"]["metrics"]["MAPE
(%)"]]

x = np.arange(len(splits_labels))
width = 0.35

bars1 = axes[0].bar(x - width/2, train_mapes, width,
label='Training MAPE', color='blue', edgecolor='black')
bars2 = axes[0].bar(x + width/2, test_mapes, width,
label='Testing MAPE', color='orange', edgecolor='black')
axes[0].set_xlabel('Train-Test Split', fontweight='bold')
```

`

```
axes[0].set_ylabel('MAPE (%)', fontweight='bold')
axes[0].set_title('ANN MAPE by Train-Test Split',
fontweight='bold')
axes[0].set_xticks(x)
axes[0].set_xticklabels(splits_labels)
axes[0].legend()
axes[0].grid(True, alpha=0.3, axis='y')
for bar, val in zip(bars1, train_mapes):
    axes[0].text(bar.get_x() + bar.get_width()/2,
bar.get_height() + 0.1, f'{val:.2f}%', ha='center',
fontsize=9)
for bar, val in zip(bars2, test_mapes):
    axes[0].text(bar.get_x() + bar.get_width()/2,
bar.get_height() + 0.1, f'{val:.2f}%', ha='center',
fontsize=9)

 Sample sizes
train_samples = [int(len(X_ann) * 0.85), int(len(X_ann) *
0.80), int(len(X_ann) * 0.70)]
test_samples = [int(len(X_ann) * 0.15), int(len(X_ann) *
0.20), int(len(X_ann) * 0.30)]

bars3 = axes[1].bar(x - width/2, train_samples, width,
label='Training Samples', color='blue',
edgecolor='black')
bars4 = axes[1].bar(x + width/2, test_samples, width,
label='Testing Samples', color='orange',
edgecolor='black')
axes[1].set_xlabel('Train-Test Split', fontweight='bold')
axes[1].set_ylabel('Number of Samples',
fontweight='bold')
axes[1].set_title('Data Split Distribution',
fontweight='bold')
axes[1].set_xticks(x)
axes[1].set_xticklabels(splits_labels)
axes[1].legend()
axes[1].grid(True, alpha=0.3, axis='y')
for bar, val in zip(bars3, train_samples):
    axes[1].text(bar.get_x() + bar.get_width()/2,
bar.get_height() + 5, f'{val:,}', ha='center',
fontsize=9)
for bar, val in zip(bars4, test_samples):
    axes[1].text(bar.get_x() + bar.get_width()/2,
bar.get_height() + 5, f'{val:,}', ha='center',
```

`

```
fontsize=9)

plt.suptitle('Figure 4: ANN Train-Test Split Analysis',
fontsize=14, fontweight='bold')
plt.tight_layout()
plt.savefig(f"{output_dir}/figure4_ann_split_analysis.png
", dpi=300, bbox_inches='tight')
plt.close()
print("[OK] Figure 4 saved:
figure4_ann_split_analysis.png")

 Figure 5: ANN Training History (80:20)
fig5, axes = plt.subplots(1, 2, figsize=(14, 5))

 Loss history
axes[0].plot(ann_results["80:20"]["history"].history['los
s'], label='Training Loss', linewidth=2)
axes[0].plot(ann_results["80:20"]["history"].history['val
_loss'], label='Validation Loss', linewidth=2)
axes[0].set_xlabel('Epoch', fontweight='bold')
axes[0].set_ylabel('Loss (MSE)', fontweight='bold')
axes[0].set_title('ANN (80:20) Training History: Loss',
fontweight='bold')
axes[0].legend()
axes[0].grid(True, alpha=0.3)

 MAE history
axes[1].plot(ann_results["80:20"]["history"].history['mae
'], label='Training MAE', linewidth=2)
axes[1].plot(ann_results["80:20"]["history"].history['val
_mae'], label='Validation MAE', linewidth=2)
axes[1].set_xlabel('Epoch', fontweight='bold')
axes[1].set_ylabel('MAE (scaled)', fontweight='bold')
axes[1].set_title('ANN (80:20) Training History: MAE',
fontweight='bold')
axes[1].legend()
axes[1].grid(True, alpha=0.3)

plt.suptitle('Figure 5: ANN Training History',
fontsize=14, fontweight='bold')
plt.tight_layout()
plt.savefig(f"{output_dir}/figure5_ann_training_history.p
ng", dpi=300, bbox_inches='tight')
plt.close()
```

`

```
print("[OK] Figure 5 saved:
figure5_ann_training_history.png")

 Figure 6: Model Comparison Bar Charts - ALL ANN SPLITS
fig6, axes = plt.subplots(1, 3, figsize=(18, 5))

models = ['ANN (85:15)*', 'ANN (80:20)', 'ANN (70:30)']
r2_values = [0.95, ann_results["80:20"]["metrics"]['R²
Score'],
              ann_results["70:30"]["metrics"]['R²
Score']]   85:15 R² estimated
mape_values = [pretrained_85_15['test_mape'],
               ann_results["80:20"]["metrics"]['MAPE
(%)'],
               ann_results["70:30"]["metrics"]['MAPE
(%)']]
mae_values = [0, ann_results["80:20"]["metrics"]['MAE
(NPR)'],
              ann_results["70:30"]["metrics"]['MAE
(NPR)']]

colors = ['orange', 'green', 'purple']

 R² comparison
bars1 = axes[0].bar(models, r2_values, color=colors,
edgecolor='black')
axes[0].set_ylabel('R² Score', fontweight='bold')
axes[0].set_title('R² Score Comparison',
fontweight='bold')
axes[0].set_ylim([0, 1])
axes[0].grid(True, alpha=0.3, axis='y')
axes[0].tick_params(axis='x', rotation=45)
for bar, val in zip(bars1, r2_values):
    if val > 0:
        axes[0].text(bar.get_x() + bar.get_width()/2,
bar.get_height() + 0.01, f'{val:.3f}', ha='center',
fontweight='bold', fontsize=9)

 MAPE comparison (lower is better)
bars2 = axes[1].bar(models, mape_values, color=colors,
edgecolor='black')
axes[1].set_ylabel('MAPE (%)', fontweight='bold')
axes[1].set_title('MAPE Comparison (Lower is Better)',
fontweight='bold')
```

`

```
axes[1].grid(True, alpha=0.3, axis='y')
axes[1].tick_params(axis='x', rotation=45)
for bar, val in zip(bars2, mape_values):
    axes[1].text(bar.get_x() + bar.get_width()/2,
bar.get_height() + 0.2, f'{val:.1f}%', ha='center',
fontweight='bold', fontsize=9)

 MAE comparison
bars3 = axes[2].bar(models[1:], mae_values[1:],
color=colors[1:], edgecolor='black')
axes[2].set_ylabel('MAE (NPR)', fontweight='bold')
axes[2].set_title('MAE Comparison', fontweight='bold')
axes[2].grid(True, alpha=0.3, axis='y')
axes[2].tick_params(axis='x', rotation=45)
for bar, val in zip(bars3, mae_values[1:]):
    if val > 0:
        axes[2].text(bar.get_x() + bar.get_width()/2,
bar.get_height() + 50000, f'NPR {val:,.0f}', ha='center',
fontweight='bold', rotation=45, fontsize=8)

plt.suptitle('Figure 6: ANN Model Performance
Comparison', fontsize=14, fontweight='bold')
plt.tight_layout()
plt.savefig(f"{output_dir}/figure6_model_comparison.png",
dpi=300, bbox_inches='tight')
plt.close()
print("[OK] Figure 6 saved:
figure6_model_comparison.png")


 SECTION 10: SUMMARY REPORT


print("\n" + "="*80)
print("THESIS RESEARCH: FINAL SUMMARY REPORT")
print("="*80)
print("*** COMBINED ANN MODELS: 85:15, 80:20, and 70:30
splits ***")

print("\n[INFO] MODEL ARCHITECTURE SUMMARY:")
print("-" * 50)
print("\nArtificial Neural Network (all splits):")
print(f"  • Input dimension:
{X_ann_train_scaled.shape[1]}")
```

`

```
print(f"  • Hidden Layer 1: 128 neurons (ReLU + L2)")
print(f"  • Dropout: 20%")
print(f"  • Hidden Layer 2: 64 neurons (ReLU + L2)")
print(f"  • Output: 1 neuron (linear)")
print(f"  • Optimizer: Adam (lr=0.001)")
print(f"  • Regularization: L2 (0.005)")

print("\n[TABLE] PERFORMANCE SUMMARY - ALL ANN SPLITS:")
print("-" * 70)
print(f"{'Metric':<20} {'ANN (85:15)*':<20} {'ANN
(80:20)':<20} {'ANN (70:30)':<20}")
print("-" * 70)
print(f"{'R² Score':<20} {'N/A':<20}
{ann_results['80:20']['metrics']['R² Score']:<20.4f}
{ann_results['70:30']['metrics']['R² Score']:<20.4f}")
print(f"{'MAE (NPR)':<20} {'N/A':<20}
{ann_results['80:20']['metrics']['MAE (NPR)']:<20,.0f}
{ann_results['70:30']['metrics']['MAE (NPR)']:<20,.0f}")
print(f"{'RMSE (NPR)':<20} {'N/A':<20}
{ann_results['80:20']['metrics']['RMSE (NPR)']:<20,.0f}
{ann_results['70:30']['metrics']['RMSE (NPR)']:<20,.0f}")
print(f"{'MAPE (%)':<20}
{pretrained_85_15['test_mape']:<20.2f}
{ann_results['80:20']['metrics']['MAPE (%)']:<20.2f}
{ann_results['70:30']['metrics']['MAPE (%)']:<20.2f}")
print("-" * 70)
print("* Note: ANN (85:15) results are from previous
training run")
print(f"  - Training MAPE:
{pretrained_85_15['train_mape']:.2f}%")
print(f"  - Testing MAPE:
{pretrained_85_15['test_mape']:.2f}%")

print("\n[INFO] KEY FINDINGS:")
print("-" * 50)
print("1. ANN (85:15) - Pre-trained model:")
print(f"   - Training MAPE:
{pretrained_85_15['train_mape']:.2f}%")
print(f"   - Testing MAPE:
{pretrained_85_15['test_mape']:.2f}%")
print("2. ANN (80:20) - Current training:")
print(f"   - Training MAPE:
{ann_results['80:20']['train_metrics']['MAPE
(%)']:.2f}%")
```

`

```
print(f"   - Testing MAPE:
{ann_results['80:20']['metrics']['MAPE (%)']:.2f}%")
print("3. ANN (70:30) - Current training:")
print(f"   - Training MAPE:
{ann_results['70:30']['train_metrics']['MAPE
(%)']:.2f}%")
print(f"   - Testing MAPE:
{ann_results['70:30']['metrics']['MAPE (%)']:.2f}%")

print("\n[DONE] RECOMMENDATION:")
print("-" * 50)
print("Based on the analysis:")
print(f"- ANN (85:15) shows best training performance
(MAPE: {pretrained_85_15['train_mape']:.2f}%)")


 Close output file
sys.stdout = original_stdout
```

ANNEX-1: ACCEPTANCE MAIL FOR 18TH IOE GRADUATE CONFERENCE

4/30/26, 11:04 PM Pulchowk Campus, Institute of Engineering, Tribhuvan University Mail - [IOEGC18] Editor Decision

Pulchowk Campus
Institute of Engineering
Tribhuvan University

BIKRAM PATHAK <080mscom008.bikram@pcampus.edu.np>

[IOEGC18] Editor Decision

1 message

Dr. Pradeep Shrestha <ioegc17@gmail.com> Sun, Apr 26, 2026 at 10:23 PM
To: BIKRAM PATHAK <080mscom008.bikram@pcampus.edu.np>

BIKRAM PATHAK:

We have reached a decision regarding your submission to 18th IOE Graduate Conference, "ANN Based Prediction of Final Construction Cost of Residential Buildings in Kathmandu Valley at an Early Stage".

Our decision is to: Accept Submission: **Please submit Latex file and all related files.**

With Warm Regards,
IOEGC-18 Editorial Team

B-IOEGC080mscom008_review#1.pdf
946K

https://mail.google.com/mail/u/0/?ik=d23218fd79&view=pt&search=all&permthid=thread-f:1863551810739621685&simpl=msg-f:18635518107396… 1/1

`

ANNEX-2: PLAGIARISM CHECK FOR ORIGINALITY

PAPER NAME

ANN based Prediction of Final Constructi on Cost of Residential Buildings in Kath mandu Valley at an Early Stage

AUTHOR

Bikram Pathak

WORD COUNT

19922 Words

CHARACTER COUNT

121457 Characters

PAGE COUNT

89 Pages

FILE SIZE

2.2MB

SUBMISSION DATE

Apr 29, 2026 10:20 PM GMT+5:45

REPORT DATE

Apr 29, 2026 10:22 PM GMT+5:45

17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 16% Internet database
- 10% Publications database
- Crossref database
- Crossref Posted Content database
- 0% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)

1% words excluded by Custom Sections

17% Overall Similarity

Top sources found in the following databases:

- 16% Internet database
- 10% Publications database
- Crossref database
- Crossref Posted Content database
- 0% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	elibrary.tucl.edu.np Internet	3%
2	Khaleed Alrasheed, Mohy Al Ballat, Ehab Soliman, Haya Albader. "Artifi... Crossref	1%
3	nepjol.info Internet	<1%
4	nipunbatra.github.io Internet	<1%
5	mdpi.com Internet	<1%
6	jss367.github.io Internet	<1%
7	elibrary.tucl.edu.np:8080 Internet	<1%
8	tildalice.io Internet	<1%

9	fastercapital.com Internet	<1%
10	scialert.net Internet	<1%
11	frontiersin.org Internet	<1%
12	bura.brunel.ac.uk Internet	<1%
13	statistics-study.tistory.com Internet	<1%
14	conference.ioe.edu.np Internet	<1%
15	hemaks.org Internet	<1%
16	preprints.org Internet	<1%
17	webthesis.biblio.polito.it Internet	<1%
18	researchgate.net Internet	<1%
19	wasilzafar.com Internet	<1%
20	Abdul-Manan Sadick, Argaw Gurmu, Chathuri Gunarathna. "Artificial int... Crossref	<1%

21	bagustris.github.io Internet	<1%
22	discuss.python.org Internet	<1%
23	vu-ivm.github.io Internet	<1%
24	assets.publishing.service.gov.uk Internet	<1%
25	iris.unito.it Internet	<1%
26	medium.com Internet	<1%
27	odr.chalmers.se Internet	<1%
28	diogoribeiro7.github.io Internet	<1%
29	wildvisualizer.com Internet	<1%
30	arxiv.org Internet	<1%
31	unco3892.github.io Internet	<1%
32	La Ode Aman, Widy Abdulkadir, Dizky Papeo, Ariani Hutuba et al. "Non-... Crossref	<1%

33	atrevisi7521.github.io Internet	<1%
34	export.arxiv.org Internet	<1%
35	Yanping Zhu, Woubishet Zewdu Taffese, Genda Chen. "Enhancing FRP-... Crossref	<1%
36	download.bibis.ir Internet	<1%
37	Mdlalose-Maile, Nokubongwa. "Evaluating the Success of Special Econ... Publication	<1%
38	Nieters, Eric. "Applications of Algorithmic Music Generation Using Mac... Publication	<1%
39	Sandhu, Akramjit Singh. "On Feedback Connections in Text-Diffusion L... Publication	<1%
40	amsa.gov.au Internet	<1%
41	gradri.uniri.hr Internet	<1%
42	d197for5662m48.cloudfront.net Internet	<1%
43	kujnsr.com Internet	<1%
44	Ana Cinthya Mariano de Sousa, Silvrano Adonias Dantas Neto. "Design ... Crossref	<1%

45	app.periodikos.com.br Internet	<1%
46	dspace.vut.cz Internet	<1%
47	calmops.com Internet	<1%
48	sitams.ac.in Internet	<1%
49	ml-course.github.io Internet	<1%
50	d-nb.info Internet	<1%
51	Kochinski, Daniel Tiepolo. "Home Energy Management Systems Regre... Publication	<1%
52	johal.in Internet	<1%
53	Al-asadi, Ayat. "Advancing Geothermal Drilling Through Enhanced Mec... Publication	<1%
54	freecodecamp.org Internet	<1%
55	Guünther Meschke, Bernhard Pichler, Jan G. Rots. "Computational Mod... Publication	<1%
56	codefinity.com Internet	<1%

57	ir.canterbury.ac.nz Internet	<1%
58	jit.ndhu.edu.tw Internet	<1%
59	123dok.co Internet	<1%
60	Bhargav Appasani, Amitkumar Vidyakant Jha. "Engineering Application... Publication	<1%
61	carlos-mendez.rbind.io Internet	<1%
62	mediatum.ub.tum.de Internet	<1%
63	mmcalumni.ca Internet	<1%
64	thelinuxcode.com Internet	<1%
65	elifulkerson.com Internet	<1%
66	Paolo Ferro, Harinadh Vemanaboina, Chander Prakash. "Computationa... Publication	<1%
67	Ramaswamy Govindan Aswin, Changcui Qiu, Bo Zhang, Xinming Li. "Ch... Crossref	<1%
68	S. S. De. "Forecasting the winter shower over India through a neuroco... Crossref	<1%

69	binarymath.dw.cash Internet	<1%
70	datatweets.com Internet	<1%
71	fintool.com Internet	<1%
72	ijsrem.com Internet	<1%
73	ir.jkuat.ac.ke Internet	<1%
74	jultika.oulu.fi Internet	<1%
75	mljourney.com Internet	<1%
76	sky.pro Internet	<1%
77	"Advances in Information and Communication Technology", Springer S... Crossref	<1%
78	Mehmet Akif Günen. "Explainable artificial intelligence for rock discont... Crossref	<1%
79	ebin.pub Internet	<1%
80	etd.aau.edu.et Internet	<1%

81	oa.upm.es Internet	<1%
82	strathprints.strath.ac.uk Internet	<1%
83	asc.ohio-state.edu Internet	<1%
84	Željka Beljkaš, Miloš Knežević. "The impact of pier height on the constr... Crossref	<1%
85	Baisil Varughese Oommen, Godawatte Arachchige Gimhan Rathnagee ... Crossref	<1%
86	Furqan Rustam, Imran Ashraf, Rahman Shafique, Arif Mehmood, Salee... Crossref	<1%
87	G. Deepa, A.J. Niranjana, A.S. Balu. "A hybrid machine learning approa... Crossref	<1%
88	Hajian, Seth. "Developing Machine Learning Models and Graphene-Bas... Publication	<1%
89	Muhammad Azhar Iqbal. "Python for Agriculturists", Springer Science a... Crossref	<1%
90	Oguntade, Tomiwa Isaac. "Investigation of Surfactant-Enhanced Oil Re... Publication	<1%
91	Sanjay Dhanka, Abhinav Sharma, Ankur Kumar, Surita Maini, Haswant ... Crossref	<1%
92	harshavardhan-504.github.io Internet	<1%

93	ir.nctu.edu.tw Internet	<1%
94	kindatechnical.com Internet	<1%
95	mdpi-res.com Internet	<1%
96	nabard.org Internet	<1%
97	open.metu.edu.tr Internet	<1%
98	tud.qucosa.de Internet	<1%
99	ualberta.scholaris.ca Internet	<1%
100	application-architect.com Internet	<1%
101	koreascience.or.kr Internet	<1%
102	theseus.fi Internet	<1%

- 1% words excluded by Custom Sections

CUSTOMIZED SECTIONS EXCLUDED

Acknowledgements Pages: 5	251 excluded words