Insurance Claim Prediction using Deep Learning for Oromia Insurance S.C

Wendimu Getachew Turu

A Thesis Submitted to Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Master's in Computer Science and Engineering (Artificial Intelligence)

Office of Graduate Studies

Adama Science and Technology University

October, 2025

Adama, Ethiopia

Insurance Claim Prediction using Deep Learning for Oromia Insurance S.C

Wendimu Getachew Turu

Advisor: Getinet Yilma, PhD

A Thesis Submitted to the Department of Computer Science and Engineering
School of Electrical Engineering and Computing
Presented in Partial Fulfillment of the Requirement for the Degree of (Artificial Intelligence) Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

October, 2025
Adama, Ethiopia

DECLARATION

I declare that this Master Thesis entitled "**Insurance Claim Prediction using Deep Learning for Oromia Insurance S.C**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Wendimu Getachew Turu	____________________	_______________
Name of student	Signature	Date

ADVISOR RECOMMENDATION

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "**Insurance Claim Prediction using Deep Learning for Oromia Insurance S.C**" prepared under my/our guidance by **Wendimu Getachew Turu** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering (Artificial Intelligences). Therefore, I/we recommend the submission of revised version of the thesis to the department following the applicable procedures.

Getinet Yilma, PhD	______________	____________
Advisor/ Supervisor	Signature	Date
________________________	________________	____________
Co-Advisor/Co-Supervisor	Signature	Date

APPROVAL PAGE

I/we, the advisors of the thesis entitled "**Insurance Claim Prediction using Deep Learning for Oromia Insurance S.C**" and developed by "**Wendimu Getachew Turu**" hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis. This is, therefore, to certify that the thesis is accepted for partial fulfilment of the requirement of the degree of Master of Science in Computer Science and Engineering (AI).

________________________	___________	_____________
Major Advisor	Signature	Date
________________________	___________	_____________
Co-advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Wendimu Getachew Turu have read and evaluated the thesis entitled "Insurance Claim Prediction using Deep Learning for Oromia Insurance S.C" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering (Artificial Intelligences).

________________________	___________	_____________
Chairperson	Signature	Date
________________________	___________	_____________
Internal Examiner	Signature	Date
________________________	___________	_____________
External Examiner	Signature	Date

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

________________________	___________	_________
Department Head	Signature	Date
________________________	___________	_________
School Dean	Signature	Date
________________________	___________	_________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENT

First and foremost, I would like to thank my God for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. This work comes in to end not only by the effort of the researcher but also the support of many individuals and organizations. To begin with, I would like to thank advisor **Getinet Yilma, PhD**, my advisor, for his constructive suggestions throughout my work. Had it been without his support, this work would not have been come in to reality.

Secondly, my heartfelt thank my wife Lelise Etana for supporting me in all of my study and moral support to achieve my work.

Thirdly, I would like to thank our colleagues/classmate friends Nesrudin Abdulwehab and Wondossen Argaw for their contribution directly and indirectly for success of my study.

Lastly, I extend my heartfelt gratitude to all the staff members of the ASTU Computer Science and Engineering department for their invaluable advice, motivation, and guidance, which have been instrumental in my first degree and master thesis research.

May God bless everyone mentioned with success and honor in their lives.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

S. C	Share Company
OIC	Oromia Insurance Company
CPU	Central Processing Unit
ML	Machine Learning
DL	Deep Learning
DNN	Deep Neural Network
RNNS	Recurrent Neural Networks
CNNS	Convolutional Neural Networks
LSTM	Long Short-Term Memory
GANs	Generative Adversarial Networks
FNN	Feed-forward neural networks
GRU	Gate Recuict Unit
RAM	Installed Memory
HMM	Hidden Markov Model
ReLU	Rectified Linear Unit
AI	Artificial Intelligence
MAE	Mean Absolute Error
MPE	Mean Percentage Error
MSE	Mean Square Error
RMSE	Root Mean Square Error
MLP	Multilayer Perceptron
GPGPU	General-purpose graphics processing unit

ABSTRACT

This study outlines the development and implementation of a deep learning model for prediction insurance claims. This study, we present a deep learning models for the task of insurance claim prediction. We compare the performance of multiple architectures, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Networks (CNN), and their hybrid combinations (LSTM-CNN, GRU-CNN, FFNN-CNN), alongside a Fully CNN model. Our findings demonstrate that hybrid models consistently outperform standalone architectures. Specifically, the LSTM-CNN model, optimized with the Adam optimizer at a learning rate of 0.001, achieved superior results across all evaluation metrics. It attained the lowest validation loss (0.0004), Mean Squared Error (0.0002), Mean Absolute Error (0.0041), Root Mean Squared Error (0.0142), and Mean Absolute Percentage Error (4.9097%), alongside the highest R^2 score (0.9977), accuracy (99.18%), and precision (99.18%). The results strongly suggest that hybrid deep learning models, particularly LSTM-CNN, coupled with the Adam optimizer, offer a highly effective and robust framework for advancing the accuracy of insurance claim predictions, thereby supporting well administrative for insurers. This study addresses the gap in Ethiopian insurance claim prediction by applying hybrid deep learning models to Oromia Insurance datasets.

KEẎWORDS

RNN, CNN-RNN, LSTM-GRU, Insurance Claim Prediction, Deep Learning

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Life is full of risks and uncertainties. Properties can be damaged by fire, personal health can be disturbed by disease, accidents can cause people to fall, products and properties can be found defective. Due to these and other factors, it is important to anticipate risks and control damages(Watt et al., 2014). As we all know risk is inevitable that is where the concept of insurance comes in. According to (Pfeffer, 1956), a technique for decreasing the risk of one party, termed the insured, through the transfer of specific risks to another party, called the insurer, who promises a restoration, at least in part, of economic losses sustained by the insured, is called insurance. A critical aspect of insurance operations is the ability to predict the likelihood of claims accurately.

Insurance is a financial term that leverages protection against potential future losses or risks(Talesh, 2015). When groups of people, organization, subscribe insurance to claim for subscription incases of future losses is an insurance. The money paid for subscription is a premium(Rejda et al., 2021a). It is an official claim made by a policyholder for their subscription to the insurance company for coverage or compensation for the loss covered or policy event. Insurance claim prediction is a critical task for insurance companies. It helps in determining the likelihood of a policyholder filing a claim, enabling insurers to make data-driven decisions about premiums, risk assessments, fraud detection, and resource allocation. Insurance providers always make a great effort, with the growing of insurance claim cost or claim loss because of insurance claim fraud(Alamir et al., 2021). Traditional statistical models have been used extensively for this purpose, but the emergence of deep learning has revolutionized the field by offering more accurate and scalable solutions. Insurance claim prediction involves forecasting whether a customer will file a claim based on historical data.

The emergence of data driven decision pushed financial institutions have implemented data collection, cleaning and storage technologies. Furthermore, artificial intelligence has a huge influence decision making (Sven Groen, 2023a). AI and ML applications true are cost-saving, improved revenue, introduced innovation in product, increased efficiency, etc. The Hidden Markov Model (HMM) has originated as a powerful tool in this regard, offering improved capabilities for modeling sequential data and uncovering latent states that influence observable outcomes(Surya et al., 2024). Generally, DNN with a single input, many hidden layers, and an output layer extensively evaluated on prediction tasks(Saputro et al., 2019). Furthermore, deep learning revolutionized the field of insurance claim prediction via capturing complex patterns from large data sources. Deep learning is making the prediction process efficient, and reliable(Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2018). It is a branch of machine learning that potentially makes neural networks to identify complex patterns in large datasets, analyze historical data to predict the likelihood of future claims by insurance holders. Specifically, Convolutional neural networks (CNN), and recurrent neural networks (RNN), have demonstrated remarkable success in various domains, such as image recognition, natural language processing, and financial forecasting(Lecun et al., 2015a).

As (Sven Groen, 2023b) pointed that, investigated predictive ML methods, such as random forest, boosting techniques, and support vector machines, which are trained for complex prediction tasks. ML have advantages over traditional statistical forecasting methods, such as moving averages and regression. ML methods leverage the capacity to capture patterns from new data points, while traditional forecasting methods mostly use predefined fixed rules generated from the data points themselves or hand-engineered hypotheses put in algorithms. ML adapts fast to the dynamics in the datasets, compared to conventional prediction methods can diminish over time. Currently, top players in the insurance corporate lack to adopt emerging technologies such as machine learning replacing traditional approaches due to many regulations associated with the insurance business and the uncertainty surrounding the new technologies(Baran & Rola, 2022; Kim et al., 2020a).

Recent developments in the area of insurance industry suggest that the emergence of deep learning, machine learning, and the application domains such as the insurance in day-to-day processes. As studied in(Burri et al., 2019a), insurance claim prediction is stated to be corner stone for market penetration in the insurance business. We have exhaustive deep and machine learning techniques are the supervised, unsupervised, and semi-supervised methods in the insurance domain(van Engelen et al., 2020; Yusuf et al., 2017). It is crucial to investigate the supervised deep learning technique for insurance data processing via deducing a function that maps a given data input as feature sets to map to data output called as targets(Abdulkadir & Fernando, 2024a).

We employ data collected from Oromia Insurance as the primary input for model training. It is important to understand the role of deep learning in improving claims prediction and in the transformation of insurance claims processing. Claims prediction is a sensitive and critical part of the insurance processing. When insurance claim data is combined with deep learning methods effectively, it not only improves the accuracy and efficiency of claim handling but also helps mitigate risk, loss, and improve profit.(Abdulkadir & Fernando, 2024b).

The proposed insurance claim prediction follows the following key sections. Firstly, the research area is defined and relevant literature is reviewed. Next, the problem is identified, and a solution is proposed, formulated, and a fitting dataset is explored to the problem definition. The dataset is then trained on the proposed model, followed by analysis and performance evaluation against existing methods. This ensures a robust, accurate, and efficient model for predicting insurance claims, and automatic processing.

The significance of applying deep learning to insurance claim prediction lies in its potential to improve efficiency and scalable claim processing. Accurate claim predictions enable insurers to assess risk more effectively, set fair premiums, and allocate resources efficiently. Despite its potential, the application of deep learning in insurance claim prediction is not without(Abdulkadir & Fernando, 2024b). Key issues include data quality, multimodal dataset, model interpretability, and computational complexity. The findings of this research have the potential to transform the insurance company by enabling more accurate risk assessment, reducing fraud, and enhancing customer experiences.

1.2 Motivation

There are several reasons why people submit insurance claims, but the main one is to safeguard their financial interests and lessen damages. Insurance claim prediction using deep learning is driven by the need to enhance profitability of insurance sector. Traditional statistical models have less capacity to handle complex and high-dimensional data, whereas deep CNNs, RNNs, and other deep learning approaches can analyze huge volumes of data amounts of structured and unstructured data, including text, images, and sensor inputs. This allows insurers to predict claims more precisely, leading to better risk assessment and pricing strategies. Accurate claim prediction is critical for maintaining profitability, minimizing fraud, and improving customer satisfaction(Singh, M., & Sharma, 2020). The motivation for adopting deep learning in insurance claim prediction stems from the need to improve performance, efficiency, and scalability in handling complex and large-scale data. Insurance companies face numerous challenges that traditional methods struggle to address effectively, making deep learning an attractive solution.

Despite these advantages, challenges like data confidentiality, model interpretability, and algorithmic bias essential be addressed to ensure principled and compliant AI deployment. Overall, deep learning transforms insurance claim prediction by improving accuracy, reducing fraud, cutting operational costs, and enabling data-driven decision-making, making it a crucial tool for modern insurers.

1.3 Statement of the Problem

It became challenging for the insurance industry to process accurate claims prediction due to the limitations in the statistical and machine learning methods. These methods are based on hypothetical rules and their low capacity to capture complex patterns from large insurance datasets. ML approaches, such as logistic regression, decision trees, and generalized linear models (GLMs), based on manual feature engineering and assumptions that are linear, which fail to capture non-linear relationships in high-dimensional insurance datasets(Hastie et al., 2009). This leads to suboptimal prediction performance, resulting in financial losses and poor risk assessment, inefficient resource allocation and claim processing, poor customer satisfaction, and missed opportunities for fraud detection (Singh, M., & Sharma, 2020). Deep learning has the potential to address these challenges successfully in insurance by leveraging its ability to automatically learn intricate patterns from diverse data types, including structured data, text, and images(Lecun et al., 2015a). According to Yunos (2016), researched on motor insurance claim processing using ANN, his work demonstrated plausible performance. There is still limited research covering other types of insurance claims. According to the work in (Burri et al., 2019a), insurance businesses' interest is to predict the future trends in insurance claims with the aim to reduce financial loss from the insurer and insured context with claim cost. As (Kim et al., 2020a) point that, proposed deep Claim which is a payer response prediction using deep learning technique. Proposed health insurance claim from patients' historical claim data modeling complex relationships from high volume of claim input (Kim et al., 2020b). Deep Claim was evaluated in its two variants and two baseline models. The deep claim score shows 23.90% relative MAE reduction whereas, the baseline score shows MAE of 6.42-day reduction on Health System A and 5.23-day reduction for Health System B)(Kim et al., 2020b). A holistic perception of deep learning model utilization for insurance claims prediction exploring activation functions in claim processing and prediction, comparing ReLU with Swish activation function concluding on the selection of the activation mechanism can have a major impact on model performance(Abdulkadir & Fernando, 2024a).

Furthermore, some research shows a considerable emphasis on using sophisticated deep learning architectures to improve the prediction of insurance claims, going beyond conventional models. For image-based tasks like vehicle damage assessment, studies by Alomar et al. (2024) and Pum (2025) effectively use Convolutional Neural Networks (CNNs) and hybrid CNN-LSTM models, attaining excellent accuracy and operating efficiency. Multi-Layer perceptron (MLPs) and sequential deep regression models have been developed for tabular data by researchers such as Al. (2022) and Abdulkadir & Fernando (2024b). Research into optimal activation functions, such as Swish, has shown performance increases over ReLU. In a similar vein, Gamaleldin et al. (2025) demonstrated remarkable accuracy in their suggested CNN-LSTM hybrid for temporal risk assessment. Common research gaps still exist, though, such as the need to handle smaller or higher-dimensional datasets more skillfully, improve model interpretability as Krishna Jampani (2019) points out, integrate unstructured data, and create thorough evaluation frameworks to close the gap between high accuracy and workable, deployable solutions for the changing insurance companies. Given the content of these research papers, a variety of insurance claim-related challenges have been covered, including Focus on single insurance claim, small dataset, Conv ML, most are single deep learning model to handle changing claim patterns and improve real-world applicability. Also, no prior study has applied hybrid deep learning models to Ethiopian insurance data.

1.4 Research Questions

When exploring insurance claim prediction using deep learning, several critical research questions can guide the investigation.

RQ1. Which variables is most significant in predicting insurance claims?

RQ2. What strategies can effectively improve the predictive power of deep learning models for insurance claims?

RQ3. Which deep learning models can improve insurance claim prediction?

1.5 Objective

The objectives of insurance claim predictions using deep learning focus on leveraging several deep learning methods to address critical challenges in the insurance industry, enhance operational efficiency, and improve decision-making.

1.5.1 General objectives

The major objective of this work is to apply a deep learning-based insurance claims prediction using structured data collected from Oromia insurance S.C.

1.5.2 Specific objectives

Specifically, the following objectives have to be addressed:

- To collect and apply several methods on the insurance claim dataset to improve quality and size;
- Apply deep learning model to improve claim prediction accuracy;
- To evaluate the performance of proposed model for Oromia Insurance S.C claim dataset.

1.6 Significance of the Study

This study aims to enhance insurance claim prediction through the use of predictive algorithms and historical claim data analysis. Studying insurance claim prediction using deep learning is importance for advancing the operational efficiency and decision-making capabilities for both insurance companies and policyholders. Insurance companies face challenges in balancing customer satisfaction, fraud detection, and cost management. The development of deep learning algorithm for insurance claims, problems with selection methods in standard machine learning, manipulation of dataset size and diversity, and application of the techniques for various insurance claim types. To improve predictive accuracy, the procedure is to integrate various sources of data into the model, such as performance measures on insurance, demographic variables, and experiential factors. Insurance claim prediction using deep learning is important because it enables insurers to make better decisions, reduce costs, improve customer satisfaction, and stay competitive in a data-driven world. It also contributes to the advancement of deep learning models in the insurance company, benefiting for business analysis and forecasting claims.

1.7 Scope and Limitations

1.7.1 Scope of the Study

While deep learning offers significant advantages in insurance claim prediction, it also has limitations and scope considerations. The insurance company’s agreements with loss of various classes of business such as motor, marine, fire and general accident, accident and health, engineering, liability, pecuniary, workmen’s compensation, micro insurance, agriculture, and political violence & terrorism. The scope of this research is to examine the potential feature in

Oromia Insurance S.C insurance claim covered all class of business, except life/health and takaful (Islamic) insurance. This research to apply deep learning procedures to predict insurance claim total payment based on their policy number recorded at branch underwriting. At the end this research, a predictive model will be created using deep learning algorithms and performance metrics.

1.7.2 Limitation of the Study

Our proposed method mainly focusses on claim processing-based insurance records collected from the insurance company. Things like the effects of public events and holidays, physical damage, and transportation are not taken into consideration in this study. The shortage of time is the main limitations in this study. Insurance firms can use deep learning to enhance their claim analysis procedures and improve results by addressing these issues.

1.8 Organization of the Study

The organization of the study is the overall content of the study which was conducted to reach the desired result. Chapter one: This part of the study is used to describe the introduction part of the study which introduces the general idea of this study like the motivation of the study, statement of the problem, research question, objective of the study, significance of study, limitation and scope, application of results, and expected outcome of the study. Chapter two: discusses the background literature and related work such as journals, articles, and magazines were reviewed in addition to the Internet. Chapter three: the research design, data collection method, data sources, data collection, data preprocessing, tools, and material used are all covered in this part. Chapter four: this section focuses on designing the architecture of the proposed model and implementation. Chapter Five: Result and discussion. Chapter Six: This is the Summary of findings, achievement of objectives, contributions, and final remarks other researchers have considered. Reference: Citations of all sources used in the study. Final appendixes: Supplementary data, code, and visualizations.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORK

2.1 Literature Review

The following section provides the current and historical insurance studies, detailing principles in insurance claims, underwriting, and claims processes. In insurance, the insurer is the company that sells insurance policies called premiums, and the insured is the person who purchases insurance to gain its benefits. The insurer pays a premium to the insured, and then the insurer takes on the insured's risk against future events or accidents(Rawat et al., 2021).

The section explores whether deep learning (DL) models can enhance the insurance industry, particularly in claim analysis to establish a broad understanding of insurance and claim predictions.

2.2 History of insurance in Ethiopia

Insurance even if in traditional form has always existed as a form of response to society problems. In the early Rome, they gathered together in burial as a support to victim family, contribute to a fund and the members of the pool had their burial costs met by the society (Birhanie, 2019). The same tradition applies for risk protection in Ethiopia using Edir to cover some financial support to losses of properties from unexpected situations. The history of modern insurance in Ethiopia dates back to the 1950s, with the establishment of the Imperial Insurance Company(Azize, 2015). The proclamation issued in march 1975 with proclamation no. 26/1975, states government ownership and control of the means of production(Azize, 2015). The banking and insurance industries were designated as government-only businesses under this declaration. Article 5 of this proclamation states that the nationalization of banks and insurance companies by the government on January 1, 1975 shall be deemed to have made under this proclamation(Zeleke, 2007). From 1974 until the late 1991, Ethiopia's insurance sector was a government control, with the Ethiopian Insurance Corporation being the only company allowed to operate (Awoke Advisor, n.d.). Following the political and economic reforms of 1991, the insurance business opened up, with the new opening of 18 local private insurance businesses. However, the insurance business has remained only domestic share companies are permitted to operate. The early roots of insurance in Ethiopia were strictly tied to expatriates and foreign insurers (Mary' & Gonfa, 2016). In fact, both expatriates and foreign businesses played a significant role in establishing the country's first domestic insurance company (Zeleke, 2007).

2.3 Over view Oromia Insurance S.C

Oromia Insurance S.C. was established, licensed, and started its operation by the National Bank of Ethiopia in 2009 with the aim of establishing insurance business. The company was started with 540 shareholders with paid-up capital of Birr 26 million. The share company later subscribed a capital of Br85 million (*Oromia Insurance - Rest. Assured.*, n.d.-b). Oromia Insurance began with a strong financial base, making it one of a kind in Ethiopia's insurance sector to collect large paid-up capital. Its broad shareholder base includes 1.6 million low-income farmers through cooperative unions. To serve these customers, Oromia Insurance Company introduced crop and livestock micro insurance in the early 2010 and launched life insurance in 2012. Now, it offers life, non-life, micro, and agricultural insurance through 62 branches, 2 contact offices, and a wide network of agents, brokers, and banks(*Oromia Insurance - Rest. Assured.*, n.d.-a).

2.4 Concept of Insurance

Insurance is a mechanism used by stakeholders as a tool for risk management, where individuals or organizations or policyholders transfer the probability of losses of liability to an insurer in exchange for a premium. Insurance is based on risk pooling, majority pay for policy, certain payments (premiums) into a common fund. This fund is then used to compensate those who experience covered losses. From an individual perspective, insurance protects by replacing uncertain large losses with a small, predictable cost. From a societal perspective, it reduces overall risk by combining many similar exposures, making losses more predictable and manageable (Rejda et al., 2021a; Zeleke, 2007). According to (Outreville, 1998), insurance is an agreement for indemnity which aim the insured to get only upto the financial loss not to make any profit. While the prerequisite is the insured pays for the premium agreed before the contingent claim is requested by the insurer as stated in the Ethiopian insurance policy, No premium No cover" policy in August 2012. According to (Outreville, 1998), insurance from the insured's point of view is a transfer of risk and from the insurer's perspective, insurance is a "pooling" mechanism of a large number of exposure units or risks. In the financial industry, insurance plays a role by directing money into various industries and supporting economic growth. Unlike most products we directly buy and use, insurance is essentially a promise we pay a fee premium, and in return the insurance company agrees to compensate you if a specific loss or damage occurs in the specific premium paid (Andrieş & Căpraru, 2014; Dickson et al., 1991).

Insurance is a mechanism of averting risk at a cost of premium (Sven Groen, 2023b). The risk is a chance, hazard, probability, and randomness in business operations, properties, and health. Therefore, insurance is a mechanism that protect individuals and or properties from suffering unpredictable risks(Spence & Zeckhauser, 1978).

2.4.1 Insurance underwriting

Insurance underwriting is the risk assessment and decision-making process used by insurers to evaluate applications for coverage, determine policy terms, and set premiums. It is the foundation of the insurance business, ensuring that risks are accurately priced and aligned with the insurer's financial stability and risk appetite(Rejda et al., 2021b). Insurance underwriting is the process by which insurers evaluate the risk associated with insuring a person, property, or asset(Angima & Mwangi, 2017). Underwriters are professionals who assess risk and establish the premium that needs to be charged to ensure that risk. The main purpose of underwriting is to ensure that insurance companies can cover potential claims without suffering financial loss. By assessing risks accurately, insurers can set premiums that are fair and reflective of the likelihood of a claim being made. Effective underwriting helps maintain the financial health of insurance companies and ensures that policyholders are charged appropriately for the level of risk they bring. Insurance underwriting bridges the gap between risk and protection, enabling insurers to operate sustainably while safeguarding policyholders from financial hardship. Advances in technology continue to refine this process, making it faster, more accurate, and inclusive. The following forms are agreement form between the Oromia Insurance S.C and insured.

2.4.2 Insurance claim

A request for coverage or payment for a covered loss or policy event made by a policyholder to their insurance provider is known as an insurance claim. For instance, if you have motor or fire insurance and you incur spare part expenses, you would submit a claim to your insurance company to get reimbursed for those costs, according to the terms of your policy.

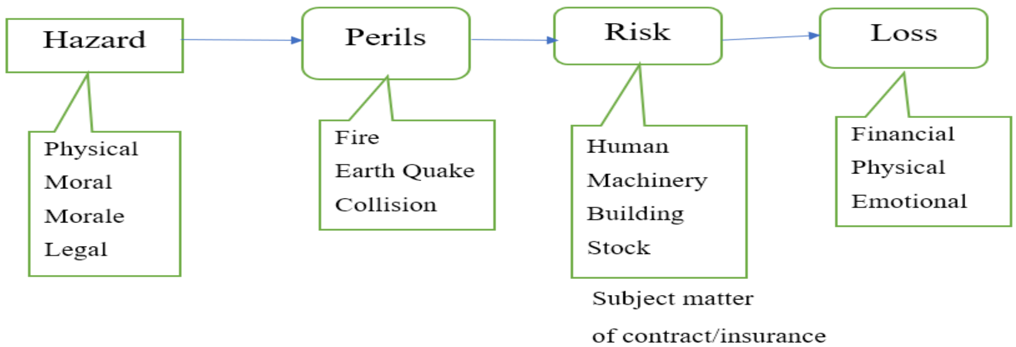


Figure 1: Step of claims occurrence

An insurance claim is a formal request made by a policyholder to an insurer for compensation or coverage following a loss or event covered under the terms of an insurance policy(Burri et al., 2019b). Claims are the cornerstone of the insurance process, enabling individuals and businesses to recover financially from unexpected incidents such as accidents, natural disasters, illnesses, or property damage(Rejda et al., 2021b).

An overview of the claim process:

- Incident Occurs: A loss or event happens that is covered under your insurance policy (e.g., a car accident, medical treatment, property damage);
- Notification: You notify your insurance company about the incident. This can often be done online, by phone, or through a claims form;
- Claim Submission: You provide necessary documentation and details about the incident. This might include police reports, medical receipts, or photos of the damage;
- Assessment: The insurance company reviews the claim, which may involve investigating the incident and evaluating the provided information;
- Approval/Denial: The insurer decides whether to approve or deny the claim based on the policy's terms and the provided evidence;
- Claim Settlement: If the claim is approved, the insurer calculates the compensation amount and pays out the claim, either directly to the policyholder or to a service provider, depending on the nature of the claim.

Insurance claims can cover a varied range of issues depending on the type of insurance, including life and health, auto, home, and marine insurance. Insurance claims can be vary depending on the type of insurance policy. In Oromia Insurance S.C there is around twelve insurance policy coverage. Insurance claims bridge the gap between risk and recovery, ensuring financial resilience for individuals and businesses.

2.4.3 Principles of Insurance

Insurance companies operate based on unique principles specifically designed for the insurance industry making every insurance contract to aims on financial security, compensation, and protection against future uncertainties. However, the insured requires act in good faith and avoid misusing this financial safeguard(Akrani, 2011). The insurance industry is built on well-defined principles that guide how contracts operate. The following are the principles

- Principle of Utmost Good Faith: Both the insurer and the insured must enter the contract with complete honesty, trust, and transparency.
- Principle of Insurable Interest: A person has an insurable interest if the existence of the insured item benefits them, while its loss would cause them financial harm.
- Principle of Indemnity: Insurance is meant to provide protection against unforeseen financial losses, not to create profit.
- Principle of Contribution: If the same subject is covered by multiple policies, the insured can only claim up to the actual loss, divided fairly among the insurers.
- Principle of Subrogation: Once the insurer compensates for a damaged or lost property, the ownership rights of that property transfer to the insurer.
- Principle of Loss Minimization: The insured must take reasonable steps to reduce or prevent further damage when a loss occurs.
- Principle of Proximate Cause: When a loss has multiple causes, the nearest and most direct cause is used to determine the insurer’s liability.

So, the insurance business relies on first, risk is transferred from an individual insured premiums to a larger group collected in the hand of the insurer, and losses are shared collectively by that group.

This system works because of the Law of Large Numbers, which ensures that by pooling risks, insurers can predict losses more accurately and spread them fairly (Akrani, 2011b; Kabeta, 2021).

2.5 New Technologies on Insurance Sector

Over the past decades, the insurance business has transformed from a slow-moving, traditional model into a dynamic, AI driven business. Once seen as too complex for high end technology driven decision making, it now inspires to apply advanced technologies related AI, blockchain, IoT, and automation for efficiency, fight fraud, improve risk assessment, and deliver personalized customer experiences. Insurers are beginning to look for innovative solutions in their insurance business, However, over half of insurers' information technology budget spent on running costs rather than research and development(Cortis et al., 2019a). Trends show all insurance businesses potentially use machine learning technologies behind their underwriting decisions over time. To mention, the buffet-style approach, where you pay the same amount irrespective of use, would not apply to motor insurance if the price gets determined with the help of machine learning technologies(Cortis et al., 2019b). The machine learning assists to accurately forecast, how much an individual is driving so that everyone pays a fair premium(Amanda & Pradipta, 2024). Also, machine learning models can be used in insurance fraud analysis, risk assessment, insurance marketing, and insurance policy pricing strategies(Sven Groen, 2023b). The wave of digitization is creating new challenges while also opening the door to tech-driven products and business models marking the rise of Insurance tech startups. Insurance tech is about integrating technology into insurance to make services smarter, and customer orientation (Rawat et al., 2021).

2.6 Deep Learning

Artificial neural networks are used in deep learning, a branch of machine learning, to model and resolve challenging issues. Deep learning, which draws inspiration from the composition and operations of the human brain, allows computers to learn from vast volumes of data without the need for explicit programming. It has revolutionized fields such as computer vision, natural language processing, speech recognition, and autonomous systems(Lecun et al., 2015a).

Deep learning, a transformative subset of machine learning (ML), leverages artificial neural networks with multiple layers known as "deep" networks to model complex patterns in data(Lecun et al., 2015b). As a pillar of modern artificial intelligence (AI), it enables machines to learn hierarchically from raw inputs, automating feature extraction and reducing the need for manual engineering. Deep learning makes it possible for computer models to gradually learn features from data at various levels. It offers solutions for practically all types of data, including text, audio, and photos. To produce outcomes in a way that is similar to what the human mind does, neural networks attempt to imitate the brain.

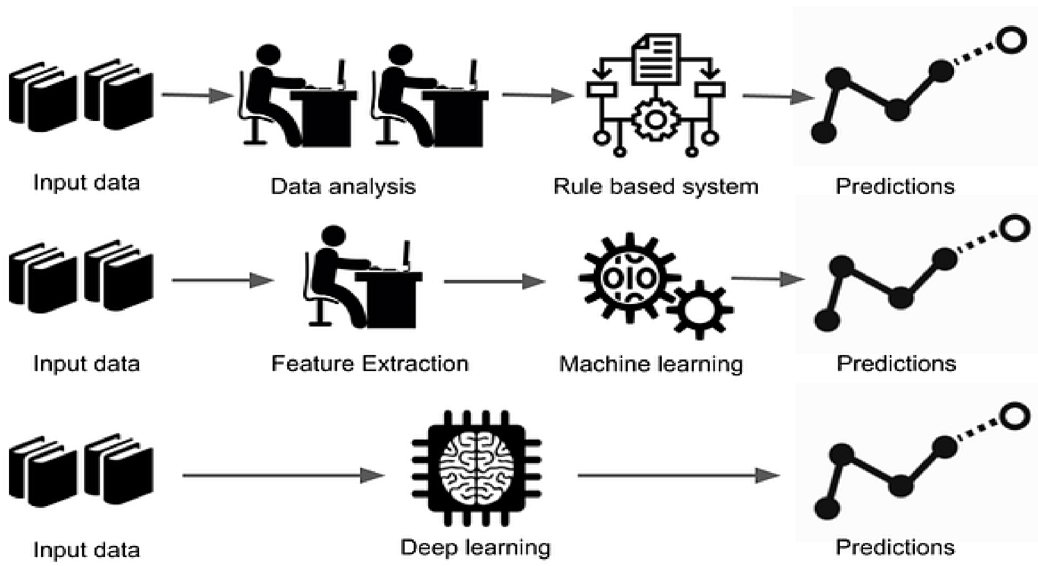


*Figure 2:Deep Learning for claim(*Accurate Insurance Claims Prediction with Deep Learning | by Lotus Labs | Medium*, n.d.).*

Each layer of neurons in deep learning models derives hierarchical features from the unprocessed input data. Deep learning automatically learns representations from unstructured data, including text, audio, and images, as contrast to traditional machine learning, which frequently necessitates human feature engineering (Bengio et al., 2021).

Multiple processing layers in computational models allow them to learn data representations at different

Deep learning allows for higher degrees of abstraction. These techniques have significantly advanced the state-of-the-art in many other fields, including drug discovery and genomics, speaker recognition, visual object recognition, object detection, and many others(Lecun et al., 2015c). Deep learning can reveal intricate structure in large data sets by using the backpropagation approach to recommend adjustments to a machine's internal parameters that are used to compute the representation in each layer based on the representation in the preceding layer.

Common structures include Generative Adversarial Networks (GANs) for image and video synthesis, Convolutional Neural Networks (CNNs) for image processing, and Recurrent Neural Networks (RNNs) and Transformers for sequential data.

The success of deep learning is largely attributed to advancements in computational power (GPUs/TPUs), availability of large-scale datasets, and improved training algorithms. Despite its impressive capabilities, challenges such as interpretability, ethical concerns, and data dependency remain critical areas of research(Schmidhuber, 2015).

2.7 Neural Network Architecture

Artificial neurons that can process several inputs and generate a single output make up neural networks, which are intricate structures. A neural network's main function is to convert input into a useful output. A neural network typically has one or more hidden layers within of its input and output layers(*Customization of Forecasting Solutions*, 2023).

In a Neural Network, all the neurons influence each other, and hence, they are all connected(Lisovsky, 2021). The network can observe every and acknowledge aspect of the dataset at hand and how the different parts of data may or may not relate to each other(Schneider & Vlachos, n.d.). This is what Neural Networks are capable of There are two methods in which information moves via a neural network:

Flow-forward networks: Signals in this paradigm only move in one direction, that of the output layer. With zero or more hidden layers, feedforward networks have a single output layer and an input layer(Islam et al., 2019). The use of pattern recognition is widespread.

Reaction Networks: In this approach, the recurrent or interactive networks process the input sequence using their internal state (memory). They have loops (hidden layer(s)) in the network that allows for signal transmission in both directions. They are frequently applied to sequential and time-series jobs.

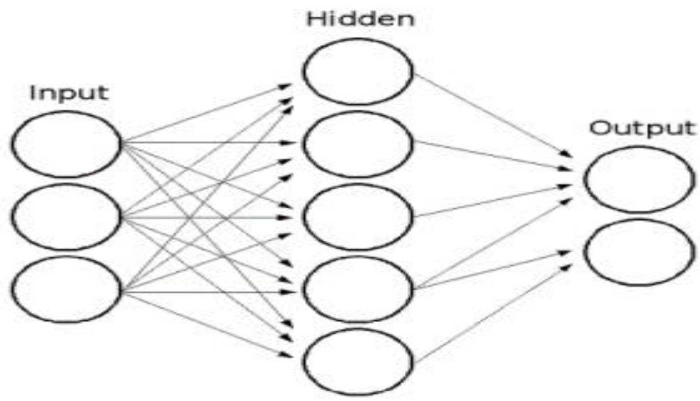


Figure 3: Components of Neural Network(Zhu & Zhang, 2016)

The learning (or training) process in a neural network is started by separating the data into three groups:

- Training dataset - With the help of this dataset, the neural network can comprehend the weights that connect its nodes.
- Validation dataset - This dataset is used to optimize how well the neural network performs.
- Test dataset - The accuracy and margin of error of the neural network are evaluated using this dataset.

Neural network algorithms are then applied to them for training the neural network after the data has been divided into these three categories. Optimization is the process of making training in a neural network easier, and the optimizer is the tool used to do it. There are various optimization algorithm types, each with particular properties and features including memory needs, numerical accuracy, and processing speed.

2.8 Deep Learning Algorithm

Deep learning algorithms will soon be integrated into a wide range of human-assistance applications. Deep learning algorithms are mimic the neuron connections of like human brain to incorporate artificial intelligence into a computer system. This increases operational speed and accuracy across a wide range of critical tasks. Computers are currently designed to perform specific tasks. The workflow procedure for deep learning algorithms across several applications is almost the same as the one depicted in the picture.4. In contrast, the neural network training procedure and dataset partitioning will be accessible for all kinds of applications. Additionally, the feature selection and preprocessing stages are not always employed. When the sample data is clean and noise-free, deep learning algorithms can be taught straight from the training dataset. Deep learning algorithms can be trained on noisy datasets. Preprocessing stages are frequently employed in the classification and prediction process to increase accuracy. In a similar vein, deep learning network training does not require the feature selection procedure. It is used in the workflow, however, to focus on some important information that is scattered across the training samples.

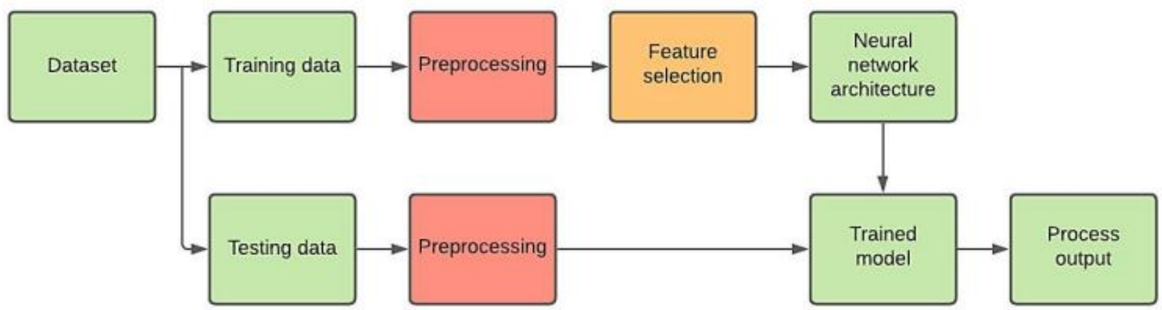


Figure 4: Work Flow of Deep Learning Algorithm

2.8.1 Deep Neural Network (DNN)

A deep neural network (DNN) employs multiple (deep) layers of highly optimized algorithms and architectures. Deep Learning, as the name implies, has more or deeper processing layers than shallow learning, which has fewer unit layers. More complicated and non-linear functions that were previously inefficiently mapped with shallow architectures can now be mapped thanks to the shift from shallow to deep learning. The widespread use of less expensive processing units, such the general-purpose graphics processing unit (GPGPU), and the abundance of training data (big data) have both contributed to this advancement. Despite having orders of magnitude more parallel processing cores than CPUs, GPGPUs are still less powerful than CPUs. As a result, GPGPUs are more appropriate for DNN implementation(Shrestha & Mahmood, 2019).

Because a single-layer network has practical limitations on the linear separable problem, the deep neural network was developed to solve any classification problem. It has one or more hidden layers, each of which contains computational nodes known as hidden nodes. The number of hidden layers in the model is its depth(Ghasemi & Miandoab, 2024). The input data is sent to the first hidden layer, which then routes its outputs to the second hidden layer, and so on. Because each layer receives the previous layer's output as an input, the input signal propagates layer by layer until it reaches the output(Nielsen, n.d.). An error signal is generated and propagated backward through the network by neurons in the output layer(Lu, 2017).

Each neuron in the hidden and output layers performs two operations. One approach is to compute the differentiable activation function on the input vectors and weights before propagating the output through the hidden layers. In the second operation, which flows backward through the network, the gradient of the error with respect to weights connected to that neuron's inputs is computed(Lu, 2017).

2.8.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a class of deep neural networks specifically designed for processing grid-like data, such as videos, images, and audio spectrograms(Hnamte & Hussain, 2023). They excel at capturing spatial hierarchies of features through convolutional operations, making them the backbone of modern computer vision systems. Convolutional Neural Networks (CNNs) are a specialized class of deep learning models designed for image processing, pattern recognition, and computer vision tasks. CNNs are very good at tasks like object detection, facial recognition, and image classification because they can effectively extract spatial hierarchies of characteristics from images(Favorskaya & Andreev, 2019). This capability eliminates the requirement for human feature extraction by enabling CNNs to learn meaningful representations straight from raw pixel data(Bengio et al., 2021).

In the field of image and handwriting recognition, convolutional neural networks are very useful. For example, they are built by first sampling an area of an image, and then utilizing the picture's characteristics to generate a representation of it. This leads to the usage of several layers, as can be seen from the description, making these models the first deep learning models(Polamuri et al., 2022).

The CNN model extracts local characteristics from the input matrix by changing the convolution kernel and then combines the local data in a higher layer to produce statistical information. A CNN is a mathematical construction that includes layers such as convolutional, pooling, and fully connected.

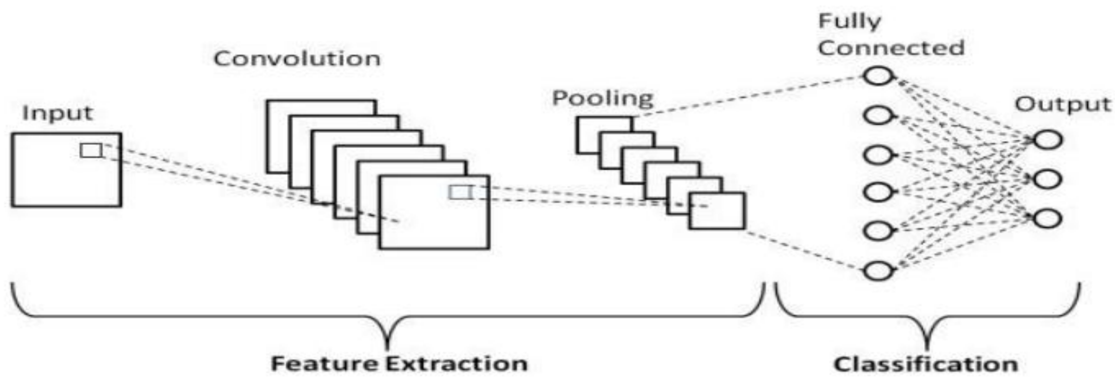


Figure 5: Architectures of Convolution Neural Network

2.8.3 Recurrent Neural Networks (RNN)

Deep learning models known as Recurrent Neural Networks (RNNs) are made to process sequential data by storing a memory of prior inputs. Because RNNs feature a feedback loop that enables them to remember information from previous time steps, they are much more effective at tasks involving time-dependent patterns than ordinary neural networks, which handle each input independently(Rumelhart et al., 1986). The key feature of RNNs is their ability to share parameters across time steps, which enables them to model temporal dependencies(Bengio et al., 2021).

A time-varying data pattern necessitates a recurrent neural network(Dinella et al., n.d.). As RNNs mature, they may be expected to unroll. In an RNN, each time step, the same layer is applied to the input (i.e., the state of previous time steps as inputs). As an input to the next firing, or time index T + 1, RNNs feature feedback loops in which the output from a previous firing or time index T is supplied. A neuron's output may be sent back to itself as an input in certain instances. RNNs come in a variety of shapes and sizes:

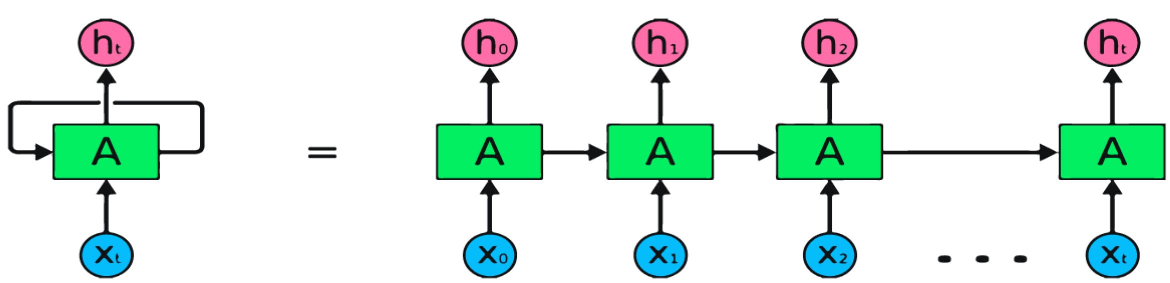


Figure 6: Recurrent neural network(Sherstinsky, n.d.)

The output at time step t and the new hidden state given the input at and hidden state of the preceding step ht−1 are calculated as:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \qquad \textit{Equation 1}$$

$$y_t = \sigma_y(W_y h_t + b_y) \qquad \textit{Equation 2}$$

- x_t is input vector a_t a time step, h_t is hidden layer vector, y_t is output vector at time step t.
- W, U, b are parameter matrices and vectors.
- σ_h, σ_y are activation functions.

Where y_t is the output sequence, σ_h, σ_y are activation functions, and W_h, U_h, b_h, W_y, by are neural network weights to be trained. Observe the relationship between the present hidden state (ht) and the prior hidden state (ht-1), which describes the characteristics of an RNN(Schmidt, n.d.).

Recurrent neural networks are constructed from various connected neural networks. They are commonly used for natural language processing in sentimental analysis and text mining. It is a link between a recurrent node in a sequence and networks with loops that assists in information persistence. In text classification, RNNs process words in a phrase recurrently and sequentially, mapping a complex and low-dimensional representation of the data into a low-dimensional vector(Soltani & Jiang, 2016). RNNs have the potential to enhance complexity and evaluate word by word, maintaining the context of texts as one of its properties. The input vector is fed into the RNN cells in the next hidden layer. Each input word has the same weight in the RNN, and the parameters are shared between different parts. RNN can only run for a limited number of steps because of problems with vanishing gradients.

2.8.4 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) designed to handle long-range dependencies in sequential data by mitigating the vanishing gradient problem(Zhao et al., 2019). In order to overcome the difficulty of learning long-term dependencies, LSTMs introduce a memory cell that stores information over longer periods of time, in contrast to typical RNNs that employ a single hidden state transferred across time (Phi, Michael, 2021). The Vanishing Gradient Problem is another name for this. An LSTM cell is distinguished by its input gate, output gate, forget gate, and cell state in addition to its hidden states. Figure 7 shows the architecture of a LSTM cell. In this figure, h_t denotes the hidden states, C_t the cell state, f_t the forget gate, i_t the input gate, and o_t the output gate(Fjellstrom, 2022). σ is the sigmoid activation function and tanh is the tanh activation function. The forget gate helps determine how much information from the previous states should be kept(Gozuoglu et al., 2024). In order to update the cell state, the input gate assists us in processing the current inputs. The sigmoid and tanh activation functions receive both the current input xt and the prior hidden state ht-1. The outputs are then multiplied by one another so that the sigmoid output assists in identifying the crucial information from the tanh output.

The mathematical relationships between the states and gates are given by

$$i_t = \sigma(x_t U^i + h_{t-1} W^i), \qquad \textit{Equation 3}$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f), \qquad \textit{Equation 4}$$

$$\sigma_t = \sigma(x_t U^o + h_{t-1} W^O), \qquad \textit{Equation 5}$$

$$C^{\sim}{}_t = \tanh(x_t U^g + h_{t-1} W^g), \qquad \textit{Equation 6}$$

$$C_t = \sigma(f_t * C_{t-1} + i_t * C^{\sim}{}_t), \qquad \textit{Equation 7}$$

$$h_t = \tanh(C_t) * o_t \ , \qquad \textit{Equation 8}$$

where U^i , U^f, U^o , W^g , W^i, W^f, W^o, W^g are trainable weights.

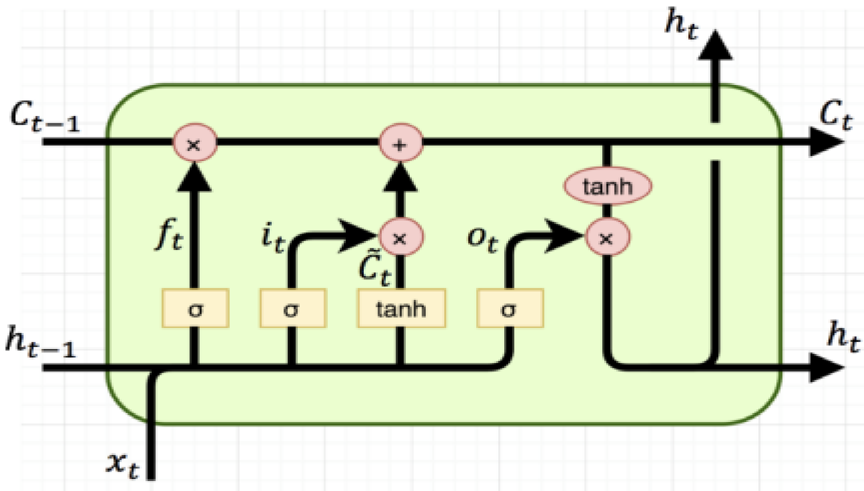


Figure 7: LSTM *architecture (Mienye et al. 2024)*

2.8.5 CNN-RNN Model

A "CNN RNN hybrid model" refers to a deep learning architecture that combines the strengths of Convolutional Neural Networks (CNNs) for extracting spatial features with Recurrent Neural Networks (RNNs) for capturing temporal dependencies within sequential data, effectively leveraging both models to achieve better performance on tasks that require analyzing both spatial and temporal information simultaneously; essentially, the CNN extracts key features from the data while the RNN understands the context and relationships within a sequence(Alomar et al., 2024).

Convolutional Neural Networks (CNNs), are mainly used for processing grid-like data such as images. They use convolutional layers to detect spatial(Islam et al., 2019). On the other hand, Recurrent Neural Networks (RNNs), are designed for sequential data like time series or text. Typically, the CNN would act as a feature extractor first. It takes the input data (like images) and extracts high-level features. Then these features are fed into the RNN, which processes the sequence. Since CNNs and RNNs are both deep learning models, the hybrid would be trained end-to-end with backpropagation. CNN-RNN hybrid model combines the strengths of both CNNs (spatial feature extraction) and RNNs (temporal sequence modeling)(Bhattacharya et al., 2024). A hybrid CNN-RNN model combines the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to handle data with both spatial (e.g., images) and temporal/sequential (e.g., time steps, text) components.

In general, the hybrid CNN-RNN leverages CNNs for spatial feature extraction and RNNs for temporal dynamics, making it powerful for tasks like video analysis, multimodal learning, and sequential decision-making.

2.9 Related work

A related work on insurance claim prediction delves into existing research and applications related to analyses in the insurance industry. It systematically analyzes published works to understand the current state of the field and identify potential areas for future development. A well conducted related work on insurance claim prediction provides valuable insights for researchers, developers, and insurance companies. It helps build upon existing knowledge and pave the way for advancements in this rapidly evolving field.

This project uses historical data from Kaggle with 1339 examples and eight variables to create a deep learning model for insurance claim prediction using sequential deep regression techniques(Abdulkadir & Fernando, 2024c). This study used a sequential model in Keras and contrasted it with activation functions that were ReLU and Swish. R squared, mean square error, and mean percentage error are the performance measures used during training to assess the model's performance. ReLU yielded values of 0.5%, 1.17%, and 23.5%, respectively, whereas the Swish function yielded values of 0.7%, 0.82%, and 21.3%(Abdulkadir & Fernando, 2024a). Developed a Multi-Layer Perceptron (MLP) model to classify claims as high-risk or low-risk. MLP achieved 92% accuracy, outperforming Decision Trees and SVM.

The purpose of this research is to provide a technique for classifying time series health data using long short-term memory (LSTM) models with a reject option the purpose of this research is to provide a technique for classifying time series health data using long short-term memory (LSTM) models with a reject option(Nam et al., 2022). We suggested a unit-wise batch standardization to address this issue, which aims to apply the structural features of LSTM models pertaining to the selection function by normalizing each hidden unit in LSTM. As a result, we verified the prediction performance of the LSTM model without holding an election. For the MIT-BIH arrhythmia data set and the human activity recognition using cellphones data set, the test accuracies of the LSTM models optimized without a selection step are 97.23% and 92.35%, respectively.

The main goal of this research effort is to use CNN models to attain the highest possible accuracy(Khinde et al., 2023). They suggest using two Convolutional Neural Network (CNN) models as part of our solution. In particular, the Mask R-CNN is used to precisely mask the damaged areas, and the VGG16 model is used to detect and evaluate the location and extent of vehicle damage(Dhokane, 2024).

In this work, we have created a Deep Claim framework for payer response prediction based on neural networks, which had not gotten much attention previously(Nam et al., 2022). By efficiently learning complex dependencies in the (high-level) claim inputs, they suggest a (low-level) context dependent compact representation of patients' past claim records. Using 2,905,026 de-identified claims data from two US health systems, we show that a deep learning-based framework called Deep Claim, built on top of this novel latent representation, can reliably anticipate different replies from diverse payers(Kim et al., 2020b). The most notable improvement in Deep Claim's ability to forecast claim rejections over properly selected baselines is a 22.21% relative recall gain (at 95% precision) on Health System A, meaning Deep Claim can detect 22.21% more denials than the optimal baseline system(Saripalli et al., 2017). The contribution of the reviewed related work is listed in Table 1.

Table 1:- Summary of related work

Author(s)	Title	Proposed Methodology	Experiment Results	Dataset size	Research Gap
(Al., 2022)	Enhancing insurance claim prediction with deep learning-based MLP models	Developed a Multi-Layer Perceptron (MLP) model to classify claims as high-risk or low-risk	Achieved 92% Decision Trees and SVM	1,338	Did not consider unstructured data (e.g., claim descriptions or images)
(Pum, 2025)	Deep Learning in Insurance Claim Automation for Car Accidents	Hybrid deep learning model combining CNNs and RNNs (LSTM)	Achieves an accuracy of 91.2%	10,000	There is a need for a more comprehensive multimodal evaluation framework for insurance claims.
(Alamir et al., 2021)	Motor Insurance Claim Status Prediction using Machine Learning Techniques	Random Forest (RF) and Multi Class Support Vector Machine (SVM).	Achieved RF 98.36% and SVM 98.17%	65,535	Focused on single claim type and machine learning model
(Navatha et al., 2023)	Forecasting Insurance Claims using Deep Learning approach	Deep learning models (e.g., neural networks)	The LSTM model 91.86% slightly better than the other models at predicting claim data.	5,56,560	Need for more cognitive models to handle evolving patterns and optimize services further.

(Gamaleldin et al., 2025)	A hybrid model based on CNN-LSTM for assessing the risk of increasing claims in insurance companies	A hybrid CNNs for spatial feature extraction and LSTM networks	TCNN-LSTM model achieved an accuracy of 98.5%, CNN (95.8%) and LSTM (92.6%) models.	2008	The gap of a more robust and accurate prediction framework that can handle the multifaceted nature of insurance data.
(Alomar et al., 2024)	A deep-learning–based antifraud system for car-insurance	Applied Convolutional Neural Networks (CNNs)	CNN accuracy 94%	Kaggle	Data expansion can be carried out to increase the size of the dataset
(Krishna Jampani, 2019)	Anomaly detection in insurance claims using deep autoencoders	Developed an autoencoder-based anomaly detection model for detecting fraudulent claims	Achieved fraud detection accuracy rates ranging from 85% to 93%	12,578	Model interpretability remains a challenge; lacks comparison with other deep learning models
(Kim et al., 2020a)	Deep Claim: Payer Response Prediction from Claims Data with Deep Learning	Deep Claim: Deep learning-based payer response prediction on de-identified claims data	The Deep Claim model predicts the response date with a relative MAE reduction of 23.9%.	2,905,026	Challenges in constructing advanced predictive analytics models
(Abdulkadir & Fernando, 2024b)	A Deep Learning Model for Insurance Claims Predictions	sequential deep regression techniques for insurance claim prediction	Better prediction both training set and test set are 0.98 and 0.71	1,339	Lowest data size, Trained data and low performance, High MSE and MPE values

2.10 Research Gaps and Challenges

The research gap analysis across the reviewed studies on insurance claim prediction using deep learning reveals several recurring challenges and opportunities for improvement. A primary gap is challenge including multimodal data integration, larger datasets, and deploy deep learning models to handle changing claim patterns and improve real-world applicability. The efficacy of deep learning models is heavily dependent on access to large volumes of high-quality, diverse data. Finally, computational complexity and resource requirements pose practical limitations. Deep learning models, especially those with complex architectures or trained on massive datasets, demand significant computational resources for both training and deployment.

CHAPTER THREE

METHODOLOGY

3.1 An Overview of the Chapter

This chapter provides an explanation of the strategies, methods, instruments, and procedures to achieve the study's objective. This section discusses a number of approaches, including feature representation techniques, data collection and preparation, data preprocessing, model design, various evaluation techniques to assess the effectiveness of the created insurance claim prediction model, and the development of tools for the suggested model.

3.2 Methodology

The phases of data collection, analysis, algorithm selection, and interpretation make up the methodology, which is used to achieve the goals and address the research issues.(Abdulwahaab, 2024. It is the scientific study of the systematic methods used in research. We discussed research methods for insurance claim prediction regarding Oromia Insurance S.C. This study examines current methodologies to extract insights from the literature, identify issues, prepare data, and evaluate our models. The diagram below illustrates the aforementioned exploration procedure.

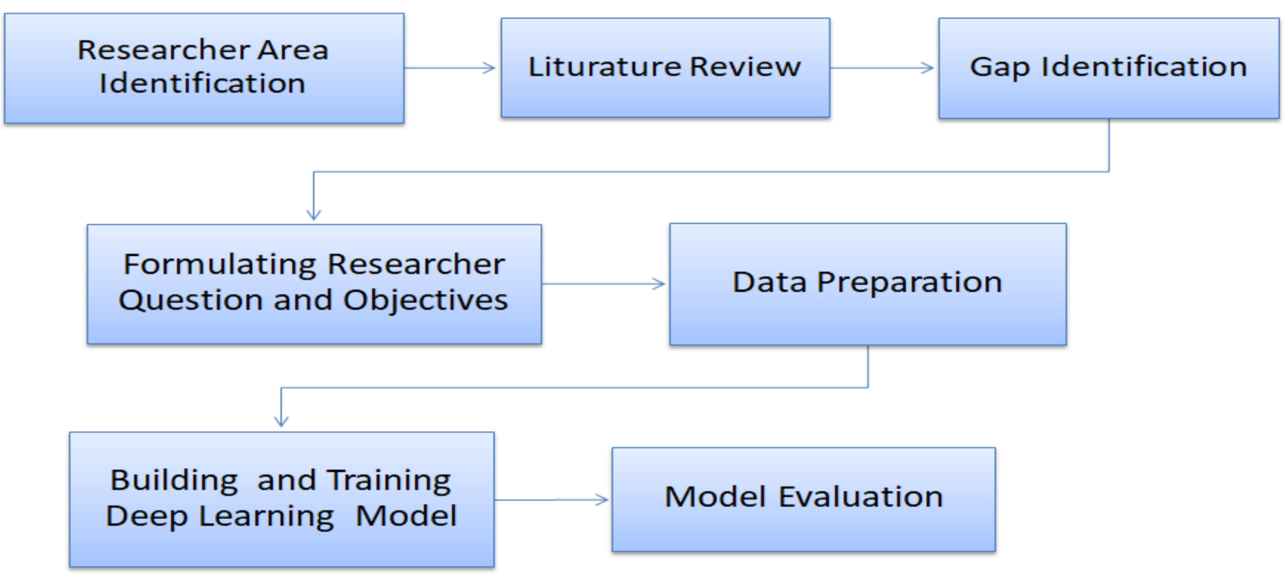


Figure 8: Research flow

Deep learning will be used in the construction of the proposed method, allowing computational models composed of many processing layers to learn data representations with different levels of abstraction. A deep-learning architecture is a multilayer stack of simple modules that are all (or almost all) topic to learning and many of which calculate non-linear input-output mappings.

3.3 Research Design

Several crucial phases are usually included in the research design for examining deep learning-based insurance claim analysis. The success of the research is greatly influenced by the choice of research design. The choices taken at this point in the research process are critical in determining the capacity of the interpretations that can be derived from the findings. The entire procedure was numerical as performance as a percentage is the expected result of this study. Thus, in order to answer the research questions outlined in Chapter 1 Research Questions 1.4, we decided to use a quantitative and experimental research design.

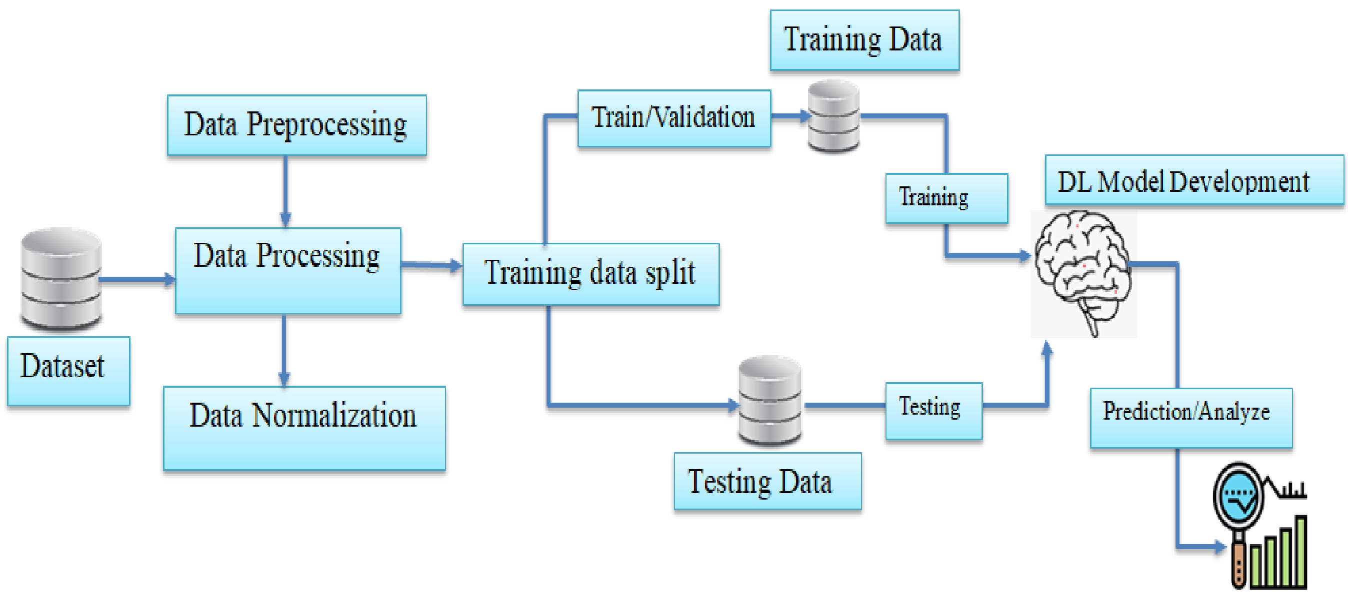


Figure 9: Proposed design model

3.4 Data Source

The dataset used in this study was collected from Oromia Insurance S.C.'s insurance claim history data. On January 26, 2009, Oromia Insurance S.C. was founded and granted a license by the National Bank of Ethiopia to concepts(*Oromia Insurance - Rest. Assured.*, n.d.-b). The historical data of the insurance claim non-life like motor, marine, accident and health, fire and general accident, engineering, liability, workmen's compensation, microinsurance, political violence &

terrorism, and pecuniary was collected and used for this insurance claim prediction. The historical claim data for thirteen years was collected and preprocessed to feed into the model.

3.5 Data collection

When it comes to machine learning or deep learning, the quality of the data is more critical than the model or algorithm. So, understanding and collecting data is an important part of machine learning technology. Even if the model is good, it will not learn anything unless the data is good and valid(Krieger et al., 2021). The mechanism for gathering and analyzing accurate data from diverse sources to feed it into a machine involves several key steps. It is one study methodology that is key to obtaining high-quality data. It is the process of gathering data from pertinent sources in order to address the research question. Whether using quantitative or qualitative methods, data gathering is necessary to ensure that the integrity of the study issue is maintained. It is a quality to collected data that determines the quality of good results, and data collection should be conducted properly. Data collection for this study will involve the acquisition of pertinent historical claim data sourced from the Oromia insurance S.C. The intended scale of the dataset is substantial, with a target of utilizing more than 32,000 datasets.

Oromia Insurance S.C. database that can assist in categorizing its clients according to their claim histories. The business task and the problem description are taken into consideration when extracting historical data on covered individuals from 2009 to 2025 from the Oromia Insurance S.C. database of insurance claims into Microsoft Excel. Relevant data was chosen in order to meet the deep learning algorithm's objective, which was derived from the claims problem.

Table 2: Claim dataset Features or attributes description

No.	Attribute Name	Data type	Description
1	S/N	Int64	Order
2	Branch	Object	Branch Code number
3	Class	Object	Types of business class
4	Product	Object	Code of business class
5	Claim no.	Object	The details of claim processed number
6	Policy no.	Object	The details of policy holder number
7	Loss of Date	Date	The date accidents occurred may be different from date of report

8	Nature of Loss	Object	Description types of loss or accident occurred
9	Name of insured	Object	Name of the Owner or Insured
10	Estimate Payment	Float32	Claim estimation payment
11	Estimate Recovery	Float32	Recovery estimation for claim payment
12	Total Estimate	Float32	Total payment of estimation payment and Recovery estimation
13	Paid Payment	Float32	Paid payment from estimation and recovery
14	Paid Recovery	Float32	Paid payment from recovery
15	Total Paid	Float32	Total payment for claim with underwriting agreement and claim procedure
16	Claim Customer Code	Object	Customer identification number or code
17	Policy Period From	Date	Starting of policy period
18	Policy Period To	Date	Ending of policy period
19	Estimate description	Object	Type of claim payment to the customers
20	Estimate Code	Number	Estimation code of type of claim payment to the customers
21	Claim Paid Date	Date	Claim paid date for customer
22	Claim Reported Date	Date	Date of accident reported from customers
23	Closed	Object	Status of registered claim

Sr_No.	Branch	Class	Product	Claim_No	Policy_No	Loss_Date	Insured_Name
1	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/17-18/1/00001	P/01/1001/17-18/1/00005	27-MAY-18	M/S.OROMIA INTERNATIONAL B
2	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/18-19/1/00001	P/01/1001/18-19/1/00002	27-NOV-18	M/S.OROMIA PUBLIC SERVICE &
3	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/18-19/1/00004	P/01/1001/18-19/1/00010	31-MAY-19	M/S.COOPERATIVE BANK OF ORO
3	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/18-19/1/00004	P/01/1001/18-19/1/00010	31-MAY-19	M/S.COOPERATIVE BANK OF ORO
4	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/18-19/1/00005	P/01/1001/17-18/1/00008	01-MAY-19	M/S.THE SECRETARIAT OFFICE OF
4	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/18-19/1/00005	P/01/1001/17-18/1/00008	01-MAY-19	M/S.THE SECRETARIAT OFFICE OF
5	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00001	P/01/1001/18-19/1/00001	28-OCT-19	M/S.OROMIA COFFEE FARMERS C
6	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00008	P/01/1001/18-19/1/00010	02-MAY-20	M/S.COOPERATIVE BANK OF ORO
6	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00008	P/01/1001/18-19/1/00010	02-MAY-20	M/S.COOPERATIVE BANK OF ORO
7	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00010	P/01/1001/17-18/1/00004	21-MAY-20	Mr.SOBOKA MULETA
7	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00010	P/01/1001/17-18/1/00004	21-MAY-20	Mr.SOBOKA MULETA
7	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00010	P/01/1001/17-18/1/00004	21-MAY-20	Mr.SOBOKA MULETA
8	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00011	P/01/1001/19-20/1/00023	28-JUN-20	M/S.OROMIA WATER WORKS CO
9	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00013	P/01/1001/17-18/1/00006	05-MAR-20	M/S.OROMIA INSURANCE COMP
10	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00014	P/01/1001/18-19/1/00011	09-SEP-19	SAT SOLAR ENGINEERING PLC
10	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00014	P/01/1001/18-19/1/00011	09-SEP-19	SAT SOLAR ENGINEERING PLC
11	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/19-20/1/00027	P/01/1001/18-19/1/00004	29-OCT-19	COOPERATIVE BANK OF OROMIA
12	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/20-21/1/00001	P/01/1001/19-20/1/00002	05-JUN-21	M/S.OROMIA NATIONAL REGION
12	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/20-21/1/00001	P/01/1001/19-20/1/00002	05-JUN-21	M/S.OROMIA NATIONAL REGION
12	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/20-21/1/00001	P/01/1001/19-20/1/00002	05-JUN-21	M/S.OROMIA NATIONAL REGION
12	01 - Finfine Branch	10 - Motor	1001 - Private Motor Car - Compre	C/01/1001/20-21/1/00001	P/01/1001/19-20/1/00002	05-JUN-21	M/S.OROMIA NATIONAL REGION

Nature_of_Loss	Estimate_Payment	Estimate_Recover	Total_Estimat	Paid_Payment	Paid_Recovery	Total_Paid	Closed	Policy_Period
Collision with TP \	3850.00	0.00	3850.00	3850.00	0.00	3850.00	N	08-OCT-17
Collision	1665.00	0.00	1665.00	1665.00	0.00	1665.00	N	18-NOV-24
Collision	1155.00	0.00	1155.00	1155.00	0.00	1155.00	N	09-FEB-21
Collision	0.00	-18288.85	-18288.85	0.00	-18288.85	-18288.85	N	09-FEB-21
Theft	20995.95	0.00	20995.95	20995.95	0.00	20995.95	N	26-MAY-18
Theft	0.00	-1500.00	-1500.00	0.00	-1500.00	-1500.00	N	26-MAY-18
Collision	745.50	0.00	745.50	745.50	0.00	745.50	N	20-MAY-19
Damage to Glasse	51875.00	0.00	51875.00	51875.00	0.00	51875.00	N	09-FEB-21
Damage to Glasse	0.00	-2593.75	-2593.75	0.00	-2593.75	-2593.75	N	09-FEB-21
Collision	14983.95	0.00	14983.95	14983.95	0.00	14983.95	Y	29-JUN-21
Collision	0.00	-1500.00	-1500.00	0.00	-1500.00	-1500.00	Y	29-JUN-21
Collision	0.00	-12097.15	-12097.15	0.00	-12097.15	-12097.15	Y	29-JUN-21
Collision with TP \	24030.45	0.00	24030.45	24030.45	0.00	24030.45	N	23-NOV-19
Collision with TP \	825.00	0.00	825.00	825.00	0.00	825.00	N	18-MAR-19
Collision	829.35	0.00	829.35	829.35	0.00	829.35	N	14-JUN-25
Collision	1760.00	0.00	1760.00	1760.00	0.00	1760.00	N	14-JUN-25
Collision	0.00	-2500.00	-2500.00	0.00	-2500.00	-2500.00	N	09-FEB-20
Collided TP Prope	340000.00	0.00	340000.00	340000.00	0.00	340000.00	N	18-JUN-21
Collided TP Prope	48695.65	0.00	48695.65	48695.65	0.00	48695.65	N	18-JUN-21
Collided TP Prope	38456.00	0.00	38456.00	38456.00	0.00	38456.00	N	18-JUN-21
Collided TP Prope	1200.00	0.00	1200.00	1200.00	0.00	1200.00	N	18-JUN-21
Collided TP Prope	0.00	-17000.00	-17000.00	0.00	-17000.00	-17000.00	N	18-JUN-21
Collided TP Prope	0.00	-456.50	-456.50	0.00	-456.50	-456.50	N	18-JUN-21
Collided TP Prope	0.00	-1913043.50	-1913043.50	0.00	-1913043.50	#########	N	18-JUN-21
Collided TP Prope	0.00	-1913043.50	-1913043.50	0.00	-1913043.50	#########	N	18-JUN-21
Collided TP Prope	0.00	1913043.50	1913043.50	0.00	1913043.50	1913043.50	N	18-JUN-21
Collided TP Prope	300.00	0.00	300.00	300.00	0.00	300.00	N	18-JUN-21
Collided TP Prope	4370.00	0.00	4370.00	4370.00	0.00	4370.00	N	18-JUN-21
Collision	620258.10	0.00	620258.10	620258.10	0.00	620258.10	N	18-MAR-23
Collision	0.00	-41010.75	-41010.75	0.00	-41010.75	-41010.75	N	18-MAR-23
Collision	1863.00	0.00	1863.00	1863.00	0.00	1863.00	N	18-MAR-23

Figure 10: Claim dataset for Oromia Insurance S.C

3.6 Data Preprocessing

In order to transform raw data which may contain incomplete, incorrect, or data type errors into a format that is suitable for a variety of uses, data preprocessing is an important step. For deep learning technology to produce the best results, reprocessing data is an essential step(W. L. Wang et al., 2019). It requires arranging the data so that its size and kind match those of the training model. In any area of artificial intelligence, well-structured data is preferred since it makes learning easier. This procedure needs standardizing the data, converting it into a supported format, and cleaning it by eliminating unnecessary information(Kingma & Ba, 2014a). Data preprocessing includes the following steps: importing the libraries, importing the dataset, identifying missing values, normalizing the data to reveal categorical values, converting categorical values to numerical values using feature selection and extraction techniques, and dividing the dataset into training and test sets. In order to improve the dataset's overall quality and similarity and set the stage for further phases of the insurance claim prediction model development, this particular preprocessing is carried out.

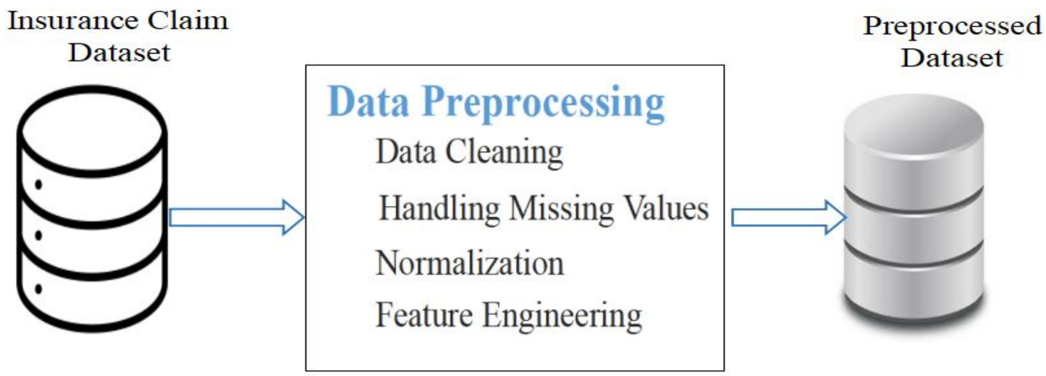


Figure 11: Preprocessing steps

3.6.1 Data Cleaning

This stage is used to ensure that there are no errors in the data. If not, several procedures such as using smoothing techniques to eliminate or reduce noise and replacing missing data with the most frequent value for that property(Witten et al., 2017). The preprocessing of modifying, fixing, and organizing data inside a data set is known as data cleaning. Data that is corrupted or unnecessary must be deleted, and it must be formatted in a language that computers can understand. Data cleansing is a method for verifying data and making sure it is good quality(Kiprotich Ng'elechei et al., 2020).

3.6.2 Handling Missing Values

This circumstance arises when some information is absent from the data. It is possible to accomplish this work in a number of conducts. Due to data collection errors and incomplete vehicle insurance claim data, several findings contained missing values. The classifier model's accuracy is impacted by the missing values. Missing values can cause major problems, such as bias in the results obtained or poor distributional representation of the entire dataset(Kang, 2013). We replace the missing values in this study using median imputation.

3.6.3 Data Normalization

Since each feature has a distinct range, feature scaling can help us have features that all contribute equally to the outcome yet have the same scale. This study employs a number of features scaling strategies, including the min-maxscaler, commonly referred to as the normalization approach. where the maximum value is changed to one and the minimum value is changed to zero. The min-max normalization formula is as follows:

$$Xn = \frac{(Xi - X\min)}{(X\max - X\min)}$$ *Equation 9*

Xn represents the normalized value of variable X.

Xi is current value of variable X;

Xmin minimum value of variable X and,

Xmax is the maximum value of variable X.

Correlation: - It is used to determine the direction and strength of the linear relationship between violations types. Using below correlation coefficient formula.

$$r = \frac{\sum (Xi - X(Yi - Y)}{\sqrt{\sum (Xi - \hat{X})2 \sum (Yt - \hat{Y})2}}$$ Equation 10

Where

Xi and Yi are individual sample point

$\overline{X}$ and $\overline{Y}$ are the mean values of the X and Y variables, respectively.

3.6.4 Feature Engineering

Feature engineering is essential for enhancing deep learning models' ability to forecast insurance claims. Developing, modifying, and choosing characteristics that improve the model's capacity to identify trends in claims data are all part of it. Improved generalization and the extraction of valuable insights from raw data are two benefits of effective feature engineering for deep learning models. For claim prediction, we will carefully identify and integrate spatial and temporal features, capturing patterns and trends that may influence claim predict activities. Additionally, we will explore interaction terms between various features to uncover relationships within dataset. In order to improve the model's capacity to identify intricate patterns and raise overall forecast accuracy, this stage attempts to extract valuable information from the raw dataset. To guarantee their applicability and efficacy in supporting the next claim prediction model, the chosen features will go through ongoing validation and improvement.

3.6.4.1 Key Feature Categories

Feature engineering for insurance claims typically involves policyholder details, claim history, financial attributes, and external risk factors. In claim variable’s, we categories into three collections: nonnumerical(Branch, Class, Product, claim no, Policy no, Insured name, Nature of Loss, Estimate Description, Closed), numerical(Sr_no, Estimate Payment, Estimate Recovery, Total Estimate, Paid Payment, Paid Recovery, Total Paid, Estimate Code) and date(Loss date Policy Period From, Policy Period To, Claim Customer Code, Claim Paid Date, Claim Reported Date).

3.6.4.2 Feature Selection

The main steps in creating deep learning models for predicting insurance claims is feature selection. The process's feature selection is another essential phase. It enhances prediction models' effectiveness and performance by concentrating on the most pertinent features and removing superfluous or irrelevant ones. The prediction process is not equally influenced by all attributes, thus choosing the appropriate features improves model accuracy, lowers computational costs, and expedites analysis. This procedure decreases overfitting, increases interpretability, and improves model accuracy(S. Wang et al., 2016).

3.7 Correlation metric

Based the dataset preprocessing and EDA identify which columns correlation to our target values. In this study we category dataset into three features (dates, numerical and non-numerical values). Numerical values with targeted values is Total paid or Total estimate with (Sr.No, Estimate paid, estimate recovery, estimate payment, recovery payment, customer code) columns, non-numerical values targeted values is Closed with (Branch, class, product, claim no, policy no, name of insured, nature of loss) columns, and dates (Loss date, policy from, policy to, claim recorded date, estimate date) columns.

3.8 Model building

In order to produce accurate prediction, the suggested deep learning-based insurance prediction system follows a defined set of procedures. Firstly, the system initiates with data collection from insurance claim, gathering crucial information regarding past claim activities. This data goes through a careful preprocessing phase, where cleaning null, normalization, up/down sampling and transformation techniques are applied to ensure consistency and quality. After that, feature selection strategies are used to find and extract the most pertinent characteristics that have a major impact on claim prediction. The dataset is then separated into training and testing sets, which establishes the framework for the model's development. To generate predictive models that allow for the prediction of potential claim activities, deep learning techniques are applied to the training data. Lastly, visualization tools are utilized to comprehensively analyze and interpret the model's results, offering insights that aid in decision-making and strategizing claim prevention measures. Through these systematic steps, the proposed system aims to leverage deep learn approach for effective claim prediction and proactive intervention strategies.

3.9 Train models

To find the best methods for our claim prediction model, we will assess a wide range of deep learning techniques during the model selection phase. The goal is to balance interpretability and prediction capability while utilizing the advantages of various methods.

3.10 Evaluation metrics

Metrics for performance evaluation are used to determine how effectively the suggested model worked with the given input data. They help us to make sure that the model performs correctly and optimally(Conrad, 2011). The metrics can be divided into two categories: classification metrics, which measure the accuracy of categorical predictions (Accuracy, Precision), and regression metrics, which evaluate continuous numerical predictions (MSE, MAE, RMSE, MAPE, R-squared)(Chicco et al., 2021). The first group concentrates on the amount and percentage of inaccuracy, whereas the second group concentrates on whether the category assignment was accurate.

Common performance metrics include R-squared, mean square error (MSE), mean percentage error (MPE), mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE). One of the most used metrics for assessing the accuracy of model predictions is mean absolute percentage error (MAPE), which is the percentage equivalent of mean absolute error (MAE). The average magnitude of mistake generated by a model, or the average deviation of predictions, is measured by mean absolute percentage error. Knowing this measure and how to compute it is vital, but when applying it in production, it's equally necessary to know its advantages and disadvantages.

$$MAPE = \frac{\sum \frac{|A-F|}{A} \times 100}{N}$$

Equation 11

Where:

N is the number of fitted points;

A is the actual value;

F is the forecast value; and

Σ is summation notation (the absolute value is summed for every forecasted point in time).

In several fields, such as banking and economic forecasting, the accuracy of predictions is assessed using a metric known as mean absolute percentage error. MAPE is commonly used as the loss function in regression problems and forecasting models due to its simple interpretation in terms of relative error for evaluation.

The Mean Absolute Error, or MAE, quantifies the average differences between expected and actual values(Chua & Huber, 2023). Formula for the mean absolute error is:

$$MAE = \frac{1}{n} x \sum_{t=1}^{n} | Yt - \hat{Y}t | \qquad \text{Equation 12}$$

Where Yt is the actual value of data point for a given time period t

$\hat{Y}t$ is the predicted values of a point for a given time period t

n is the total numbers of data points

In continuous motion prediction, root means square error, or RMSE, is frequently employed as a performance metric(D. K. Sharma et al., 2022). It determines the average difference between the actual data points and the projected values to avoid canceling out positive and negative values when they are added up. When comparing models, this measure makes significant errors appear more prominent. To choose the optimal model, RMSE is computed using de-normalized input and forecasted output data. The equation for calculating RMSE:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (Yt - \hat{Y}t)^2}{n}} \qquad \text{Equation 13}$$

Where Yt is the actual value of data point for a given time period t

$\hat{Y}t$ are the predicted values of a point for a given time period t

n is the total numbers of data points

While root mean square error (RMSE) measures the discrepancy between expected and actual values, standard deviation measures the distribution of data around the mean. The degree of dispersion of these residuals is measured by the RMS(Chai & Draxler, 2014). Stated differently, it indicates the degree of concentration of the data around the line of best fit(Bulmer, 2024).

Models using Swish activation function have shown better performance metrics compared to those using ReLU. The rectified linear unit (ReLU), also known as the rectifier activation function, resolves the vanishing gradients problem and adds the nonlinearity property to a deep learning model. It interprets the argument's strong point(Dubey & Jain, 2019a).

$$f(x) = \max(0, x)$$ *Equation 14*

Compared to its predecessor activation functions like sigmoid or tanh, it is straightforward yet significantly superior. The ReLU function is monotonic, as is its derivative.

An important technique for assessing how well binary classification algorithms work is the AUC-ROC curve. Plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at different levels shows how well a model can distinguish between two groups, such as positive and negative outcomes. AUC-ROC provides an overall performance score over all possible categorization levels. It takes into account the model's performance at various operational points, in contrast to accuracy, precision, or F1-score, which are dependent on a particular threshold(Fawcett, 2006; Majnik & Bosnić, 2013).

Important AUC-ROC terms include:

- TPR (True Positive Rate), which is the proportion of accurately predicted positive cases.
- FPR (False Positive Rate): The proportion of negative cases that were m is predicted.
- Specificity: The percentage of real negatives that the model accurately recognized (inverse of FPR).
- Sensitivity/Recall: The percentage of true positives that the model properly detected (also known as TPR).

3.11 Materials

Insurance claim prediction requires a variety of materials, including software tools, programming languages, development frameworks, and data sources.

3.11.1 Hardware Working Environments

It uses a laptop that has the following specifications. Processor 13th Gen Intel(R) Core(TM) i5-1334U, 1300 MHz, 10 Cores, 12 Logical Processors, 8 GB of physical memory, 512 GB SSD storage, 64 GB of flash, additional external drives, and Microsoft Windows 10 Enterprise, 64-bit computer operating system.

3.11.2 Software implementation environments

Table 3: Packages and tools used during the implementation

Packages and tools used	Descriptions
Jupyter Notebook's	Is helpful for machine learning, modeling, data visualization, and data transformation and cleansing.
Matplotlib's	Python publication quality metrics. It is used in this study to visualize the data and findings.
Numpy's	processing arrays for objects, strings, and integers. It is used in this work to manage the process of turning text into numerical data for features, model testing, and training.
Panda's	Pandas is a helpful package that offers quick, adaptable, and expressive data structures for handling "labeled" or "relational" data.
Python	Python is a well-liked and simple-to-learn programming language that may be used to develop machine learning applications.
Sklearn	A Python module for machine learning that manages the fundamental ML tasks of classification, regression, and clustering.
Seaborns	Visualization of statistical data
Keras	Python-based, high-level neural network API that is simple to use and operates on top of TensorFlow. To allude to the model construction's testing and training
TensorFlow	The most widely utilized and well-liked deep learning framework. TensorFlow offers several advantages, including improved efficiency when working with multidimensional measurements in scientific expressions.
Google Colab's	Is a cloud-based notebook from Google that enables you to develop Python code on any browser with sufficient system resources and training power.
Microsoft	used to prepare documents
Mendeley desktop	Used for handling reference manager

CHAPTER FOUR

PROPOSED SOLUTION AND IMPLEMENTATION

4.1 Chapter Overview

This chapter covers the suggested approach to claim insurance prediction using deep learning algorithms that are appropriate for resolving the identified problem by preparing a claim insurance dataset according to the model, data types, and columns or characteristics. This section will explain various points such as the general architecture of the designed model, proposed data preprocessing, proposed feature extraction techniques, proposed deep learning model, implementation of data preprocessing, implementation of deep learning methods, and proposed model implementation. The research's dataset was gathered from Oromia Insurance S.C., and using the methodology covered in chapter three, we then prepared the dataset for prediction.

4.2 Problem Identification

This study aims to create a deep learning model that can make predictions using a dataset of insurance claim metrics. The prediction task involves predicting safety label of the insurance claim based on various recorded claim behavior. This study leverages a deep learning techniques to capture insurance claim in the data that traditional machine learning model might miss. The dataset consists of several claim metrics, including date of report, claim no, policy no, insured's name, date of accident, loss estimate, paid, recovery estimate, total estimate, total paid, and status.

4.3 Proposed Insurance Claim Prediction Model

The prediction model was constructed using deep learning algorithms after the related features for insurance claim prediction were identified and validated, as well as after the claim's history dataset was gathered. The main objective of this project is to develop an insurance claim prediction model using CNN, RNN, and LSTM in order to identify which deep learning model is most suited for claim prediction. The model does not use the collected dataset directly. After being filtered or altered, it was produced in a format suitable for the deep learning model. The dataset is preprocessed to make the methodology comprehensive and capable when the model receives it as input. Data cleansing, resolving missing values, and handling feature categorization are all part of the first step in the dataset preprocessing process. In order to choose relevant features, feature engineering selection uses the following feature selection techniques after preprocessing the dataset: sequential feature, recursive feature elimination, embedded approaches, and feature selection by merging multiple. Third, split dataset as training, validation and testing. Four, train deep learning models like LSTM, CNN-RNN hybrids, and RNN. We use CNN for complex spatial context, RNN for long large context. Fifthly, apply the evaluation model methods to overcome performance of output. The proposed architecture of this study serves as framework for implementing the model. For model development, we use open source, and free data science libraries. IDEs including Sklearn, Seaborn, Matlabs, Jupyter, Spyder, Keras, Tensflow, and Notebook are all included with Anaconda Navigator. We utilize Keras to build our model and conduct tests.

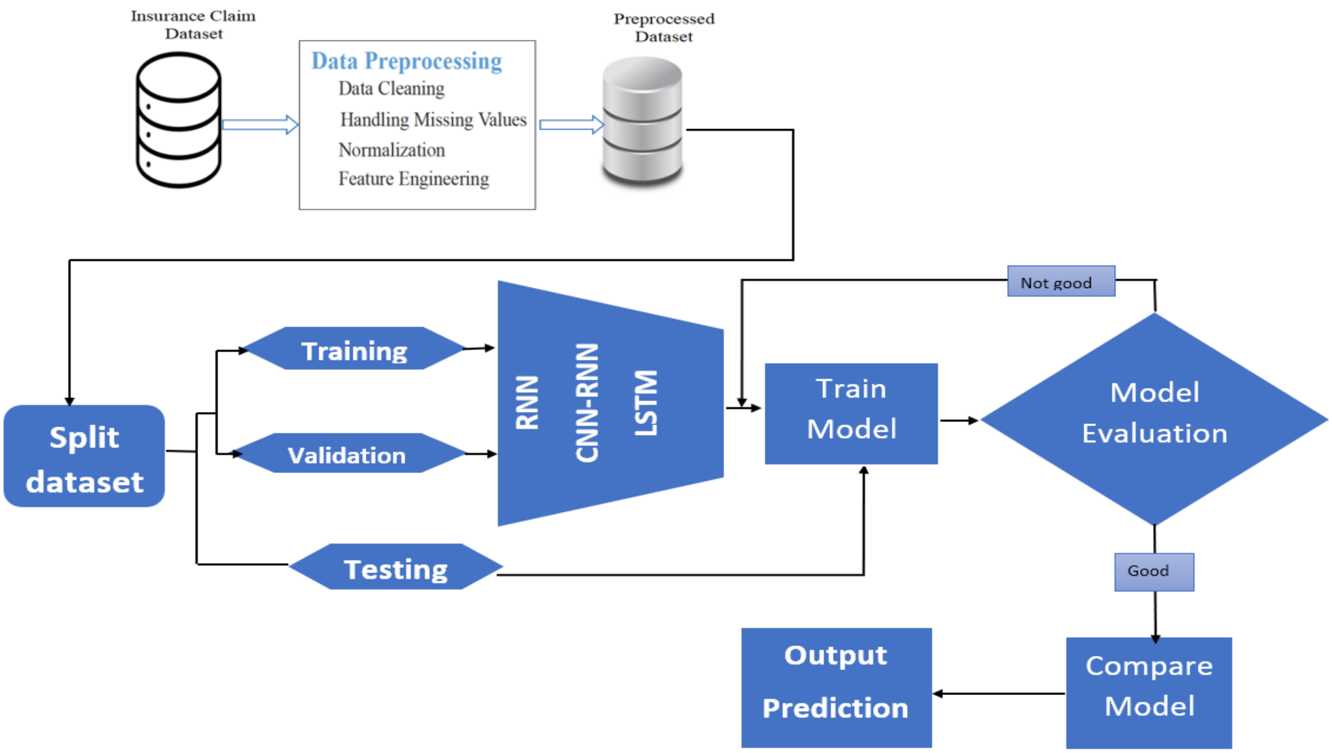


Figure 12: Prediction Model Architecture

4.4 Dataset Descriptions

To construct the dataset for this research, we manually gathered and documented insurance claim history data from Oromia Insurance S.C. The known properties or features were taken into consideration when gathering the insurance claim history data from 2009 to 2025 in order to prepare the dataset. The dataset comprises 32391 rows and 23 features/columns of data, and through the use of data preprocessing techniques, missing value handling, data cleaning, misbalancing, data normalization, and feature engineering are accomplished using various feature selection techniques such as sequential, recursive, and embedded methods. The dataset is used as input for the suggested approach, which is then preprocessed to make it comprehensive(Asresa et al., 2023).

4.5 Variables and Data Set

Among the variables considered in this investigation were Sr_no, Branch, Class, Product, claim no, Policy no, Loss date, Insured name, Nature of Loss, Estimate Payment, Estimate Recovery, Total Estimate, Paid Payment, Paid Recovery, Total Paid, Closed, Policy Period From, Policy Period To, Claim Customer Code, Claim Paid Date, Estimate Code, Estimate Description, Claim Reported Date. Table 2 displays the feature descriptions.

The raw data was gathered from Oromia Insurance S.C. in order to construct the claim predictions. The resulting dataset included 32391 rows and 23 columns. Out of the 23 variables, we 22 as input and one (Total paid) as target value. Using 80:20:10 random states and Sklearn's train_test_split function, the dataset was split into trains, validation, and tests. In Figure 9, the dataset's structure is displayed.

4.6 Deep Learning Model Building

Because of its ability to model intricate patterns, deep learning has undergone significant development over time, radically changing a number of fields. Deep learning (DL), which performs very well in a variety of applications and exhibits strong capabilities in managing enormous volumes of data and intricate calculations, has dramatically changed the field of artificial intelligence (AI)(O'Halloran et al., 2024). This study's model was created using neural network models, including CNN, RNN, and LSTM. CNNs, which were first created for image processing jobs, provide ground-breaking results in fields like image and video recognition by using layers of convolution operations to record spatial hierarchies in data(Mienye & Jere, 2024a). RNNs, on the other hand, were created to process sequence data and have uses in anything from language modeling to speech recognition(Sherstinsky, 2020). By avoiding the vanishing gradient issue that plagued previous iterations, the addition of LSTM units improved the model's efficacy by addressing issues(Mienye & Jere, 2024b).

Neural networks, on the other hand, are frequently divided into shallow and deep learning. The number of hidden layers in the network is a key factor that separates shallow and deep learning models. While deep neural networks with several hidden layers can extract hierarchical features and create more complicated data representations, shallow networks, which usually have one or two hidden layers, are restricted in their capacity to learn complex features(Mienye & Swart, 2024a). The neural network receives inputs and processes them in hidden layers to produce predictions as an output. During training, these layers' weights are changed. After the models are constructed, appropriate evaluation criteria are used to determine how well the models forecast insurance claims. The model with the lowest loss function and the highest accuracy is selected as the final prediction insurance claim model. The sequential deep regression model, a neural network library, and Keras were used in the study.

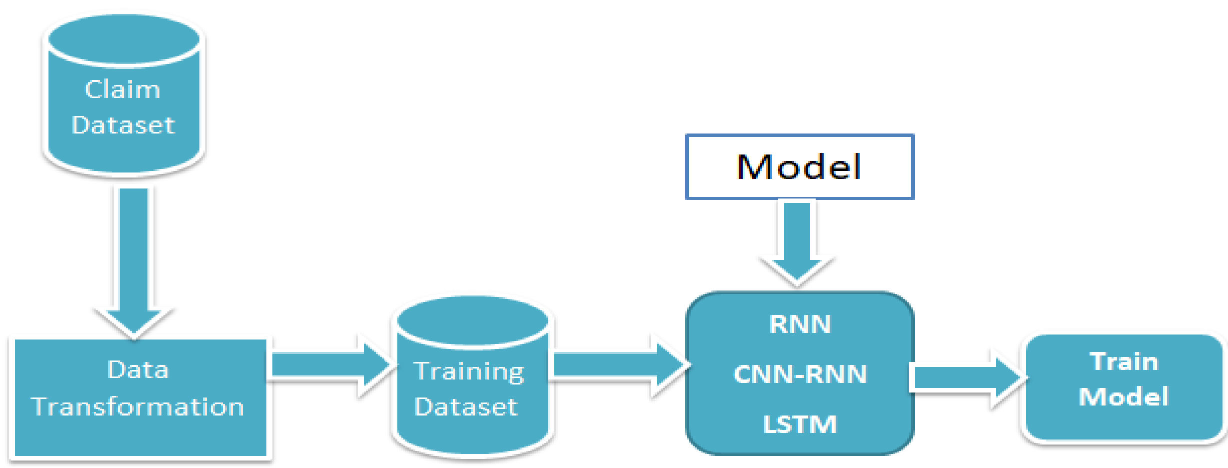


Figure 13: Model Building Flow Diagram

4.7 Components of Deep Learning Models

Deep learning models can learn from data, make predictions, and optimize performance thanks to their fundamental building blocks. Each of these components' layers, optimization techniques, activation functions, and loss functions are required for the model to learn from and generalize from data(Mienye & Swart, 2024b).

4.7.1 Layers

The basic units of deep learning models are layers, which dictate how information is processed and changed as it passes through the network. A typical deep learning model consists of an input layer, one or more hidden layers, and an output layer(Mienye, Swart, et al., 2024). Raw data, like text or photos, is sent to the input layer and then passes through the hidden layers for different calculations and transformations. The model learns to extract appropriate features and patterns from the data in the hidden layers. Lastly, using the information that has been analyzed, the output layer generates the final classification or prediction. Dense or completely connected layers, which are usually located near the end of a network, enable high-level reasoning by connecting each neuron to every other neuron in the layer above. This is done by using the features that the layers before it has retrieved(Mienye, Sun, et al., 2024).

Applications requiring time series or natural language processing might benefit from recurrent layers, such as those found in RNNs, and their variations, such as LSTMs and gated recurrent units (GRUs), which are made to capture temporal dependencies in sequential data(Mienye, Swart, et al., 2024). By accurately identifying and expressing complicated patterns in the data, these specific layers allow deep learning models to manage a wide variety of challenging jobs.

4.7.2 Activation Functions

Deep learning models require activation functions because they provide nonlinearity into the network, which helps it recognize intricate patterns and generate accurate predictions. A neural network's capacity to represent intricate relationships in the data would be limited in the absence of activation functions, which would only allow it to carry out linear transformations(S. Sharma et al., 2020). Common activation functions that have different functions within a model include the sigmoid function, softmax function, and Rectified Linear Unit (ReLU). When dealing with binary classification situations where the results are understood as probabilities, the sigmoid function is helpful because it converts input values to the range (0, 1)(Enyinna Nwankpa et al., 2018). Sigmoid purposes, on the other hand, may be less successful in deep networks due to disappearing gradients.

$$\sigma(x) = {}^{1}\!/_{1} + \ell - x$$ *Equation 15*

“This problem is addressed by ReLU, which is defined as ReLU(x) = max(0, x), which outputs the input directly if it is positive and zero otherwise. ReLU is the most often used activation function in deep learning models' hidden layers because it is easy to use and efficient at resolving the vanishing gradient issue, which enables quicker (Dubey & Jain, 2019) converg(Dubey & Jain, 2019b)019b). Moreover, the representation of the softmax function is:

$$soft\max(zi) = \frac{\ell^{zi}}{\sum_{j}\ell^{zj}}$$ *Equation 16*

The denominator normalizes the output to a probability distribution across all potential classes, while Zi stands for the logit for class I. It is usually applied to multi-class classification issues at the output layer of a neural network. In order to interpret the model's predictions as class probabilities, it transforms the raw model outputs (logits) into probabilities that add up to one(Enyinna Nwankpa et al., 2018).

4.7.3 Loss Functions

Loss functions, sometimes referred to as cost functions or objective functions, are essential parts of deep learning model training because they calculate the discrepancy between the actual target values and the model's predictions. By suggesting how the model parameters should be changed to reduce mistakes and increase predicted accuracy, loss functions process(Mienye & Swart, 2024b). Because it has a direct impact on how the model learns during training, the loss function selection is contingent upon the specific task and dataset features.

The Mean Squared Error (MSE) is a popular loss function that is mostly employed in regression problems. MSE determines the average squared difference between the model's predicted values Þi and the actual target values ŷi:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Equation 17

where n is the number of samples, Y_i is the true value, and $\hat{Y}_i$ is the projected value. MSE is vulnerable to outliers since it penalizes greater errors more harshly due to the squaring term. When major mistakes need to be drastically decreased, this sensitivity is helpful, but it can also cause problems if the dataset include outliers that distort the loss function.

4.7.4 Optimization Algorithms

Deep learning relies heavily on optimization methods since they minimize the loss function by modifying the model's parameters(Mienye & Swart, 2024b). Since it directly affects the convergence rate and solution quality, effective optimization is crucial for deep learning model training. To improve training efficacy and performance, a variety of optimization techniques have been developed, each with advantages and disadvantages. Stochastic Gradient Descent: One of the fundamental optimization techniques in deep learning is stochastic gradient descent(Mienye, Sun, et al., 2024; Mienye & Swart, 2024c). By adjusting the learning rate to the parameters, the Adaptive

Gradient Algorithm (AdaGrad) makes smaller updates for often occurring parameters and bigger updates for infrequent ones(Liu et al., 2019). By maintaining a moving average of the squared gradients, Root Mean Square Propagation (RMSProp), an adaptive learning rate technique, overcomes Adam learning rates(Mienye & Swart, 2024b). This permits more constant training and prevents the learning rate from being too low. A more complex optimization algorithm called the Adaptive Moment Estimation (Adam) optimizer combines the benefits of two additional SGD extensions: RMSProp, which scales the learning rate based on a moving average of recent gradients, and AdaGrad, which modifies the learning rate for each parameter balance gradients(Kingma & Ba, 2014b).

4.8 Loading of Dataset

The dataset is loaded from the file location directory where each subdirectory represents a different class of insurance claim files. To load the dataset, the pandas, seaborn, library was used to read CSV files representing the collective dataset for insurance claim prediction. The CSV files were loaded into pandas with data frames.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Try loading with different encodings
try:
    df = pd.read_csv('Claim_data.csv', encoding='latin-1')
    print("Dataset loaded successfully with latin-1 encoding!")
except UnicodeDecodeError:
    try:
        df = pd.read_csv('Claim_data.csv', encoding='windows-1252')
        print("Dataset loaded successfully with windows-1252 encoding!")
    except UnicodeDecodeError:
        print("Error: Could not decode the file with UTF-8, latin-1, or windows-1252 encoding. Try a different encoding or inspect t
        exit()
    except FileNotFoundError:
        print("Error: Claim_data.csv not found. Please make sure the file is in the correct directory.")
        exit()
except FileNotFoundError:
    print("Error: Claim_data.csv not found. Please make sure the file is in the correct directory.")
    exit()

df.info()
```

Figure 14: Loading dataset, information and description

4.8.1 Data Visualization

Visualizing the processes of understanding the dataset and classifying any class inequalities or null values in data.

```python
# Visualize null value counts for all columns
plt.figure(figsize=(12, 6))
sns.barplot(x=null_counts.index, y=null_counts.values)
plt.title('Number of Null Values per Column')
plt.xlabel('Column Name')
plt.ylabel('Number of Null Values')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
Null value counts per column:
Sr_no.              0
Branch              0
Class               0
Product             0
Claim_no            0
Policy_no           0
Loss_date           0
Insured_name        0
```

Figure 15: Dataset visualization

4.8.2 Feature Scaling (Normalization)

This step of implementation is used to convert data in different range features into the same range between 0 and 1. This method is applied only to the column with numeric values in the insurance claim dataset. In LSTM and RNN if training data is normalized, the model reduces the neural network from weighting feature inequalities. This study used MinMaxScaler and imputation library from sklearn's to transform the given dataset.

4.8.3 Correlation metric

The correlation is used to classify based the dataset preprocessing and EDA which columns correlation to our target values. The correlation of dataset (Sr_no, Branch, Class, Product, claim no, Policy no, Loss date, Insured name, Nature of Loss, Estimate Payment, Estimate Recovery, Total Estimate, Paid Payment, Paid Recovery, Total Paid, Closed, Policy Period From, Policy Period To, Claim Customer Code, Claim Paid Date, Estimate Code, Estimate Description, Claim Reported Date) used in this study is:

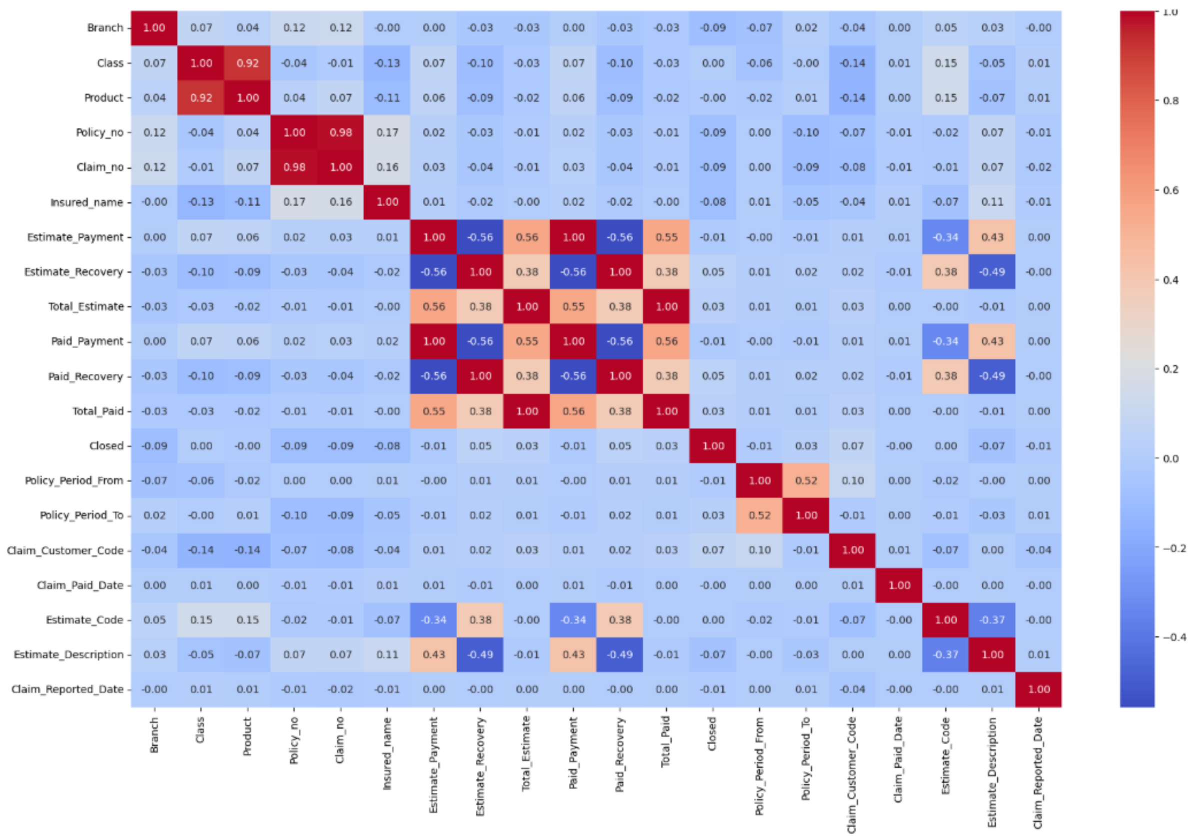


Figure 16: Correlation metrics

4.8.4 Data Splitting

To verify that the model is trained on a subset of the data and validated and tested on distinct subsets, the dataset is divided into training, validation, and test sets. With 32391 divisions into train, validation, and testing (80:20:10), the prepossessed dataset has 23 rows total, of which 22 are classified as feature values and one as target values.

4.8.5 Engineering Features

In order to understand how well deep learning models perform in predicting insurance claims, feature engineering is essential. It requires generating, transforming, and selecting characteristics that improve the model's capacity to identify trends in the claim's dataset. Efficient feature creation helps deep learning models enhance generalization and uncover valuable insights from raw datasets. In order to determine which outputs will perform the best, this technique classifies the feature selection from the raw dataset for the suggested model. It entails the innovative process of creating and developing new features that more effectively capture the underlying relationships and patterns in the data.

4.9 Implementation of Model Training

One of the most important phases in creating a reliable identification for insurance claim prediction is the model training and assessment procedure. The next stage involves developing the deep learning algorithms and training the model using the training dataset following the preprocessing and loading of the dataset. Creating, training, and assessing are the next phases after data preprocessing. As discussed in section 4.3 this is a study proposed with Keras Models API. Next, two dimensions, X and Y, are created using the train test and validation data. The first 22 columns are labeled X Train, Y Train, and X validation; the final column is labeled Y Train, Y Validation, and Y Test.

The implementation of the six models is shown in this section. Our goal in putting these six models into practice was to categorize the best strategy for predicting insurance claims. Each model's performance rating would be determined in part by the analysis and findings of its evaluation.

4.9.1 Implementation of CNN model

CNN is a highly performing model especially in the area of computer vision. Recently many researchers are using CNN for different areas including speaker recognition. TensorFlow's deep learning framework is used to create the CNN model for predictive maintenance. With the help of high per parameters that resulted good performance the code model is built sample bellow:

```
# Function to create 2-layer CNN model
def create_model(input_shape):
    model = Sequential()
    model.add(Input(shape=input_shape))
    model.add(Conv1D(filters=32, kernel_size=3, activation='relu', padding='same'))
    model.add(Conv1D(filters=16, kernel_size=3, activation='relu', padding='same'))
    model.add(Flatten())
    model.add(Dense(50, activation='relu'))
    model.add(Dense(1))
    return model
```

Figure 17: 2-Layer CNN model code implementations

Important patterns and features from the input time series data are captured by the model's various convolutional layers, which have different filter sizes, kernel sizes, and activation functions. The spatial dimensions are subsequently decreased by feeding the convolutional layers' output into a global average pooling layer. Lastly, 64 generates the binary predictions by adding a dense output layer with a sigmoid activation function. The model's parameters include the number of epochs, in/output channel size, kernel size, batch size, and evaluation metrics including R-squared, MSE, MAE, MAPE, and RMSE. It is compiled using the Adam optimizer, RMSprop, and SGD.

4.9.2 Implementation of RNN model

Insurance claim prediction is a good fit for Recurrent Neural Networks (RNNs), a potent deep learning architecture intended to handle sequential data. For time-sensitive insurance datasets, recurrent neural networks (RNNs), especially Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) variants, provide a reliable solution by processing sequential data and modeling long-term relationships.

```
# Build RNN model
model = tf.keras.Sequential([
    tf.keras.layers.LSTM(64, return_sequences=True, input_shape=(1, X_train.shape[2])),
    tf.keras.layers.LSTM(128, return_sequences=True),
    tf.keras.layers.LSTM(256),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1)
])

model.compile(optimizer='adam',
              loss='mse',
              metrics=[
                  tf.keras.metrics.RootMeanSquaredError(),
                  tf.keras.metrics.MeanAbsoluteError(),
                  tf.keras.metrics.MeanAbsolutePercentageError(),
                  r_squared
              ])
```

Figure 18: Implementation code of RNN

4.9.3 Implementation hybrid of RNN and CNN model

By utilizing the advantages of both architectures, a hybrid strategy that combines recurrent neural networks (RNNs) with convolutional neural networks (CNNs) provides a potent answer. Because RNNs (particularly LSTMs or GRUs) excel at processing sequential data, they are perfect for identifying time-dependent patterns in insurance claims. When examining structured data, CNNs can be helpful in identifying patterns and spatial features. RNN layers are used to capture temporal dependencies after CNN layers have extracted spatial data. Use measures like as R-squared, MSE, MAE, MAPE, and RMSE to evaluate the performance of the model.

```python
# Hybrid CNN-LSTM Architecture
model = tf.keras.Sequential([
    # CNN Feature Extraction
    tf.keras.layers.Conv1D(64, 3, activation='relu', input_shape=(X_train.shape[1], 1)),
    tf.keras.layers.MaxPooling1D(2),
    tf.keras.layers.Conv1D(128, 3, activation='relu'),
    tf.keras.layers.MaxPooling1D(2),

    # LSTM Sequence Processing
    tf.keras.layers.Reshape((-1, 128)),  # Convert CNN output to sequence
    tf.keras.layers.LSTM(128, return_sequences=True),
    tf.keras.layers.LSTM(64),

    # Regression Output
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(1)
])
```

Figure 19: Hybrids CNN-RNN implementations code

4.9.4 Implementation of LSTM model

An artificial neural network called long short-term memory (LSTM) is employed in deep learning and artificial intelligence. The model is constructed as demonstrated in the example code below with the aid of high parameters that produced good performance:

```python
# Function to create 2-layer LSTM model
def create_model(input_shape):
    model = Sequential()
    model.add(Input(shape=input_shape))
    model.add(LSTM(50, activation='relu', return_sequences=True))
    model.add(LSTM(50, activation='relu'))
    model.add(Dense(1))
    return model
```

Figure 20: 2-layer LSTM model code Implementations.

4.10 Compile the Model

The model is constructed and trained using the MSE loss and the Adam optimizer, which is a type of stochastic gradient descent. By modifying the training parameters, the network will try to minimize log loss during training (weights). Optimizer is used to modify the parameter gradients after they have been calculated using back-propagation.

4.11 Model Training

The process of fitting the deep learning model with the optimal weights and biases to minimize a loss function across a prediction range is known as model training. The model will be trained after the network is constructed. The suggested number of iterations for model training is 50, though this could be altered. The code below was then used to train the model.

4.12 Model Testing and Evaluation

A range of evaluation indicators, which are enumerated and explained in section 3.3, were used to gauge the model's performance. R-squared, root mean square error, mean absolute error (MAE), mean square error (MSE), mean percentage error (MPE), and mean absolute percentage error (MAPE), (RMSE) are instances of typical performance metrics.

```python
# Plot training curves
plt.figure(figsize=(12, 6))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('RNN Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('MSE Loss')
plt.legend()
plt.show()

# Evaluate model
y_pred = model.predict(X_test)
print("\nTest Metrics:")
print(f'MSE: {mean_squared_error(y_test, y_pred):.4f}')
print(f'RMSE: {np.sqrt(mean_squared_error(y_test, y_pred)):.4f}')
print(f'MAE: {mean_absolute_error(y_test, y_pred):.4f}')
print(f'MAPE: {np.mean(np.abs((y_test - y_pred.squeeze()) / y_test)) * 100:.2f}%')
print(f'R²: {r2_score(y_test, y_pred):.4f}')
```

Figure 21: Evaluation metrics

4.13 Hyperparameter Tuning

To maximize a neural network model's performance, hyperparameter adjustment is required. In this study, we experimented with various hyperparameters, focusing on the number of epochs and the architecture of convolutional layers. Performs hyperparameter tuning by testing three optimizers (Adam, RMSprop, SGD) with two learning rates each (Adam/RMSprop: 0.001, 0.0001; SGD: 0.01, 0.001), selecting the model with the lowest validation loss.

CHAPTER FIVE
RESULTS AND DISCUSSION

5.1 Chapter Overview

In addition to the results of several evaluation metrics, such as R-squared, mean square error (MSE), mean percentage error (MPE), mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE), this section discusses the findings of multiple experiments using different model architectures and data balancing strategies. After the previous discussion of the implementation concepts and details of this work, this chapter examines the results of the suggested solution and offers a comparative description of the outcomes using graphs and tables.

5.2 Model Building

Six models FCNN, FFNN, RNN, CNN, RNN-CNN, and LSTM were used in this investigation. First, the data set is separated into groups that are independent and dependent. The dependent feature is the goal or class whose value depends on the independent features. Features whose values are not dictated by others are known as independent features. The models were then constructed using the binary class datasets.

5.3 Model Evaluation Results

We used our preprocessed dataset in our study to evaluate the performance of our suggested models, and we calculated model evaluation metrics to determine the models' efficacy after testing and training them using performance evaluation metrics covered in chapters three and four.

5.4 Findings for the Deep Learning Models

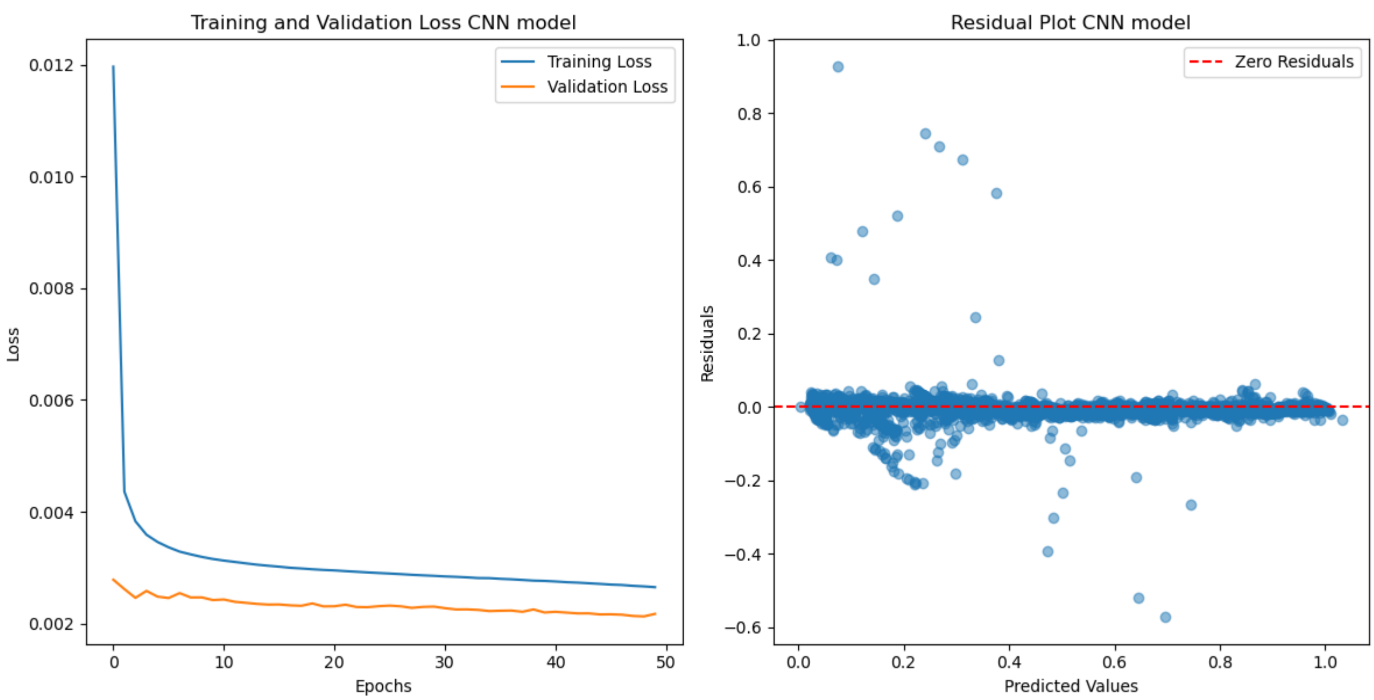
5.4.1 CNN Model

CNN's model was evaluated across various optimizers and learning rates, producing different final validation losses.

Table 4: 2-Layer CNN Training Model

Epochs	Batch size	Activation	Optimizer	Learning rate	Validation losses	Best Optimizer
50	128	relu and dense 1	Adam	0.001	0.0024	
			Adam	0.0001	0.0023	

			RMSprop	0.001	0.0022	RMSprop with Learning Rate 0.001 Best Validation Loss: 0.0022
			RMSprop	0.0001	0.0023	
			SGD	0.01	0.0025	
			SGD	0.001	0.0030	



```
Test Set Metrics (Best CNN Model):
MSE: 0.0019
MAE: 0.0160
RMSE: 0.0441
MAPE: 21.0991
R2: 0.9780
Accuracy: 96.8161
Precision: 96.8161

Sample Predictions (10 points):
Actual vs Predicted Total_Paid:
Sample 1: Actual=0.22, Predicted=0.21
Sample 2: Actual=0.22, Predicted=0.22
Sample 3: Actual=0.89, Predicted=0.89
Sample 4: Actual=0.92, Predicted=0.92
Sample 5: Actual=0.50, Predicted=0.51
Sample 6: Actual=0.96, Predicted=0.96
Sample 7: Actual=0.79, Predicted=0.79
Sample 8: Actual=0.81, Predicted=0.83
Sample 9: Actual=0.23, Predicted=0.22
Sample 10: Actual=0.47, Predicted=0.48
```

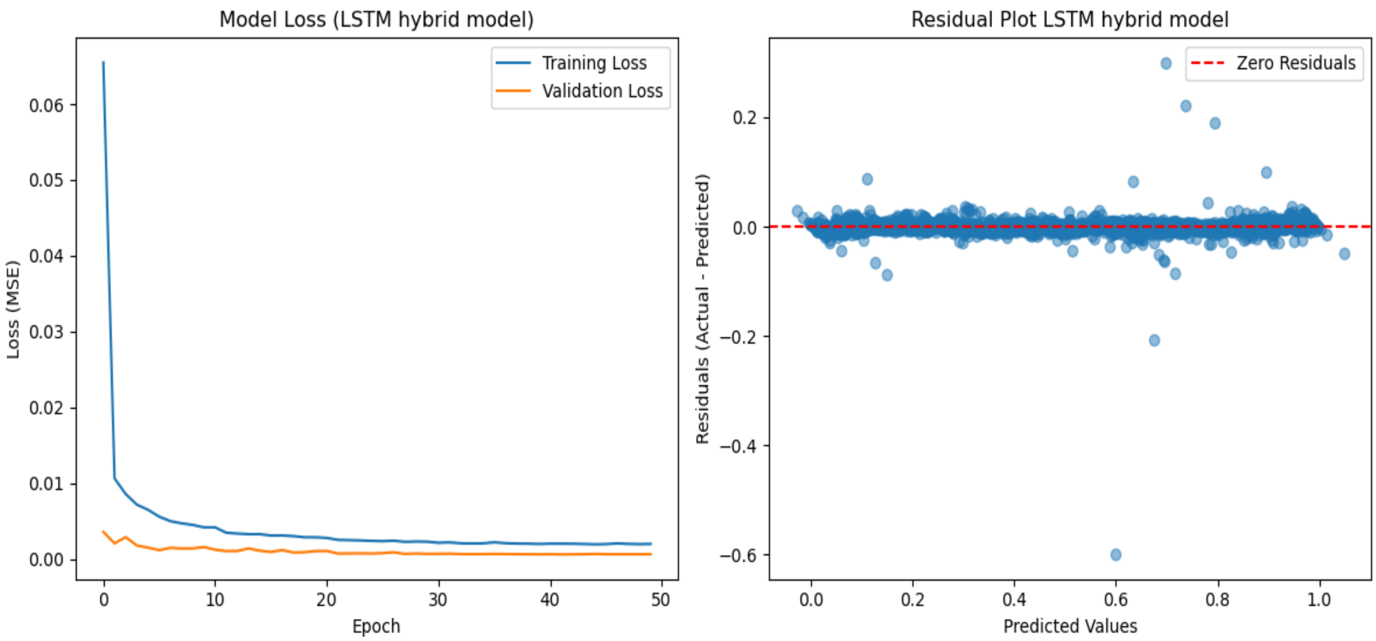
Figure 22: 2-Layer CNN Training Model output

5.4.2 LSTM Model

Evaluation of the LSTM model was conducted different optimizers and learning rates, resulting in varying final validation losses.

Table 5: 2-Layer LSTM Training Model

Epochs	Batch size	Activation	Optimizer	Learning rate	Validation losses	Best Optimizer
50	128	relu and dense 1	Adam	0.001	0.0007	Adam with Learning Rate: 0.001 Best Validation Loss: 0.0007
			Adam	0.0001	0.0026	
			RMSprop	0.001	0.0024	
			RMSprop	0.0001	0.0027	
			SGD	0.01	0.0072	
			SGD	0.001	0.0798	




```
Test Set Metrics LSTM hybrid model:
MSE: 0.0002
MAE: 0.0058
RMSE: 0.0157
MAPE: 5.3168
R2: 0.9972
Accuracy: 98.8399
Precision: 98.8399

Sample Predictions (10 points):
Actual vs Predicted Total_Paid:
Sample 1: Actual=0.08, Predicted=0.07
Sample 2: Actual=0.89, Predicted=0.88
Sample 3: Actual=0.36, Predicted=0.36
Sample 4: Actual=0.50, Predicted=0.51
Sample 5: Actual=0.10, Predicted=0.10
Sample 6: Actual=0.24, Predicted=0.23
Sample 7: Actual=0.59, Predicted=0.58
Sample 8: Actual=0.06, Predicted=0.06
Sample 9: Actual=0.66, Predicted=0.66
Sample 10: Actual=0.72, Predicted=0.72
```

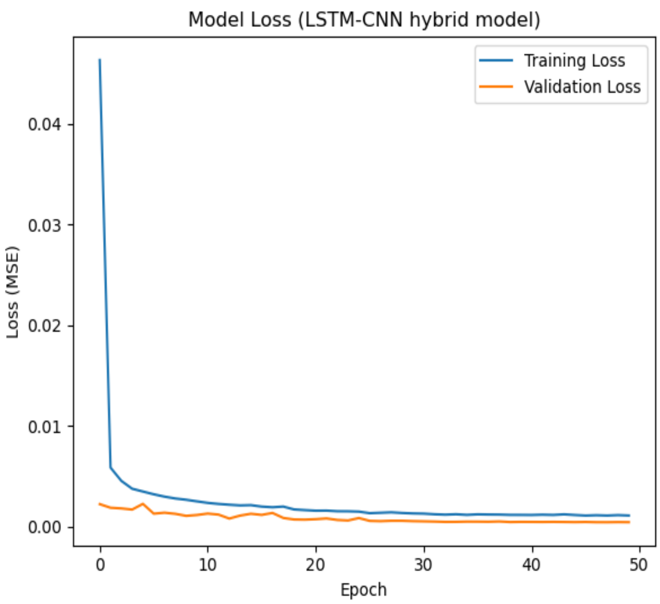
Figure 23: 2-Layer LSTM Training Model output

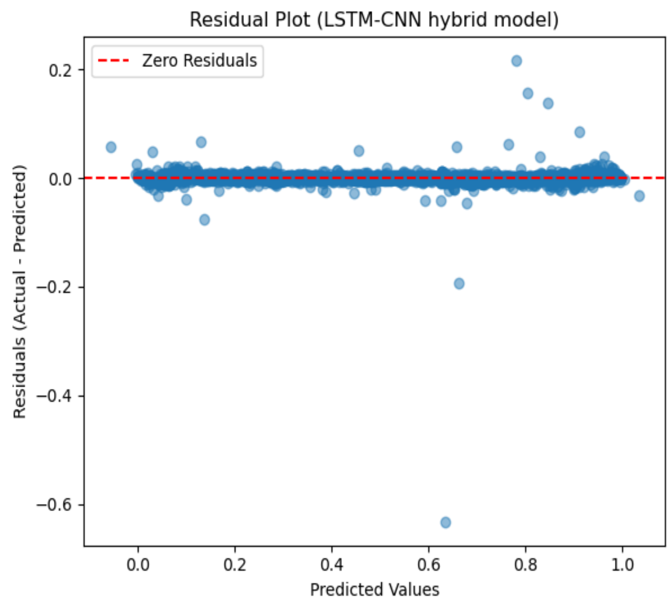
5.4.3 LSTM-CNN Model

The different optimizers and learning rates were using to evaluated LSTM-CNN model, resulting in varied final validation losses.

Table 6: 2-Layer LSTM-CNN Training Model

Epochs	Batch size	Activation	Optimizer	Learning rate	Validation losses	Best Optimizer
50	128	relu and dense 1	Adam	0.001	0.0023	Adam with Learning Rate: 0.001 Best Validation Loss: 0.0004
			Adam	0.0001	0.0024	
			RMSprop	0.001	0.0025	
			RMSprop	0.0001	0.0025	
			SGD	0.01	0.0026	
			SGD	0.001	0.0517	





```
Test Set Metrics (LSTM-CNN hybrid model):
MSE: 0.0002
MAE: 0.0041
RMSE: 0.0142
MAPE: 4.9097
R2: 0.9977
Accuracy: 99.1767
Precision: 99.1767

Sample Predictions (10 points):
Actual vs Predicted Total_Paid:
Sample 1: Actual=0.08, Predicted=0.07
Sample 2: Actual=0.89, Predicted=0.88
Sample 3: Actual=0.36, Predicted=0.35
Sample 4: Actual=0.50, Predicted=0.50
Sample 5: Actual=0.10, Predicted=0.10
Sample 6: Actual=0.24, Predicted=0.24
Sample 7: Actual=0.59, Predicted=0.59
Sample 8: Actual=0.06, Predicted=0.06
Sample 9: Actual=0.66, Predicted=0.67
Sample 10: Actual=0.72, Predicted=0.73
```

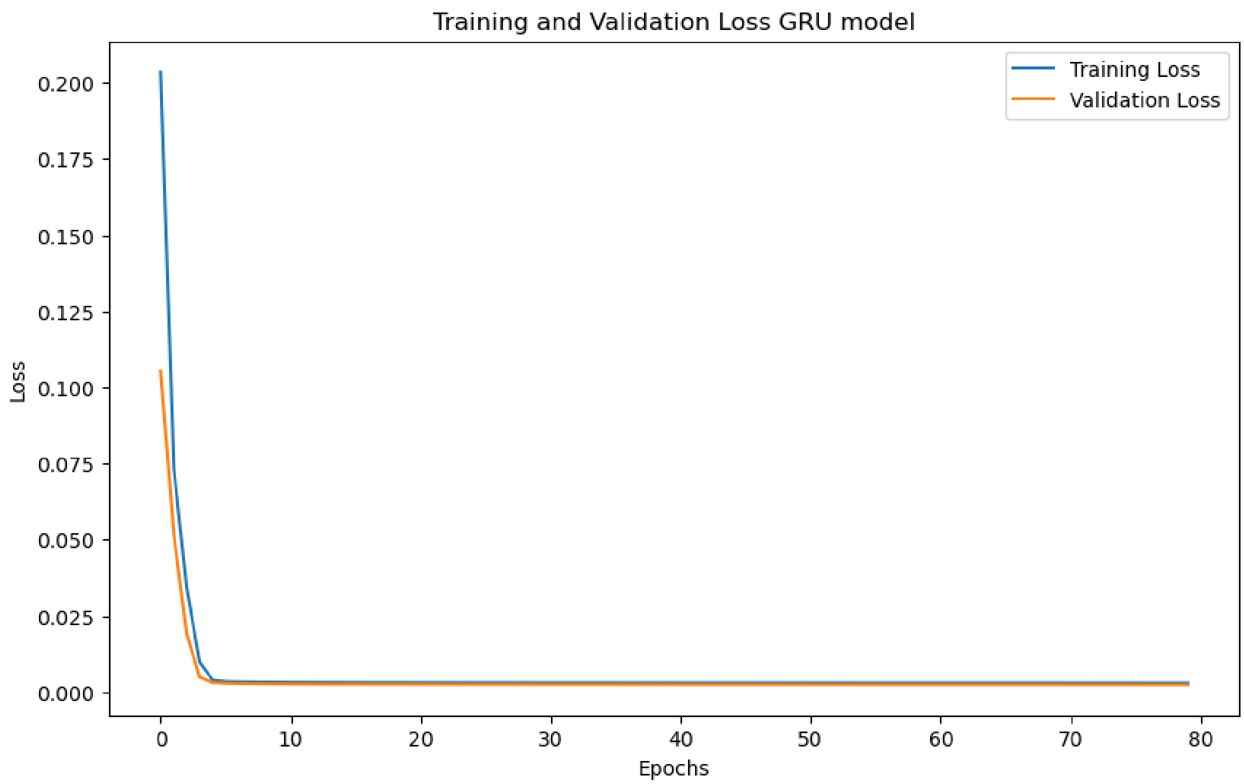
Figure 24: 2-Layer LSTM-CNN Training Model output

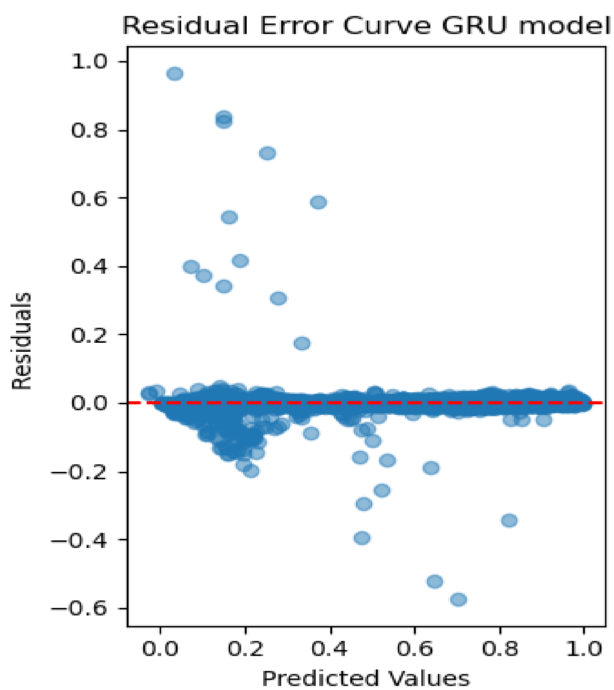
5.4.4 GRU Model

Deferent optimizers and learning rates were used to assess the GRU model, resulting in different final validation losses.

Table 7: 2-Layer GRU Training Model

Epochs	Batch size	Activation	Optimizer	Learning rate	Validation losses	Best Optimizer
50	128	relu and dense 1	Adam	0.001	0.0024	Adam with Learning Rate 0.0001 Best Validation Loss: 0.0024
			Adam	0.0001	0.0024	
			RMSprop	0.001	0.0024	
			RMSprop	0.0001	0.0025	
			SGD	0.01	0.0026	
			SGD	0.001	0.0335	






```
Test Set Metrics (Best GRU model):
MSE: 0.0021
MAE: 0.0132
RMSE: 0.0453
MAPE: 17.4852
R2: 0.9767
Accuracy: 97.3750
Precision: 97.3750

Sample Predictions (10 points):
Actual vs Predicted Total_Paid:
Sample 1: Actual=0.22, Predicted=0.22
Sample 2: Actual=0.22, Predicted=0.25
Sample 3: Actual=0.89, Predicted=0.90
Sample 4: Actual=0.92, Predicted=0.92
Sample 5: Actual=0.50, Predicted=0.50
Sample 6: Actual=0.96, Predicted=0.96
Sample 7: Actual=0.79, Predicted=0.79
Sample 8: Actual=0.81, Predicted=0.81
Sample 9: Actual=0.23, Predicted=0.23
```

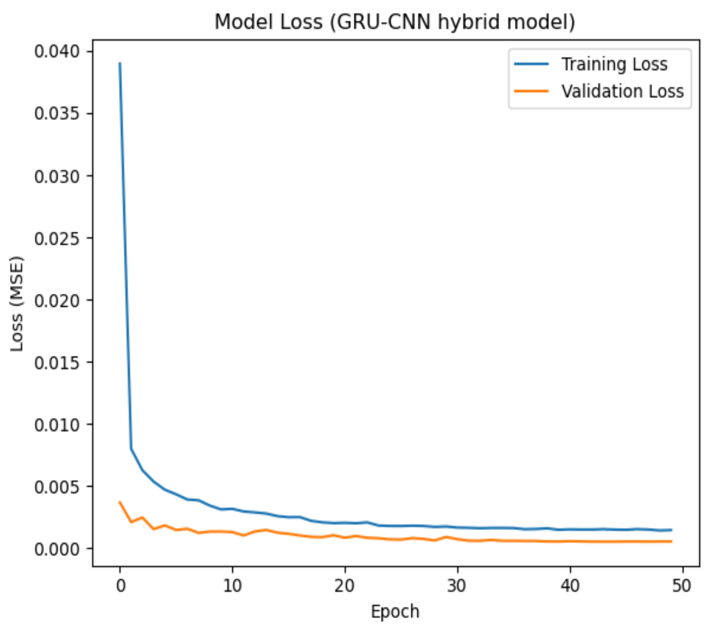
Figure 25: 2-Layer GRU Training Model output

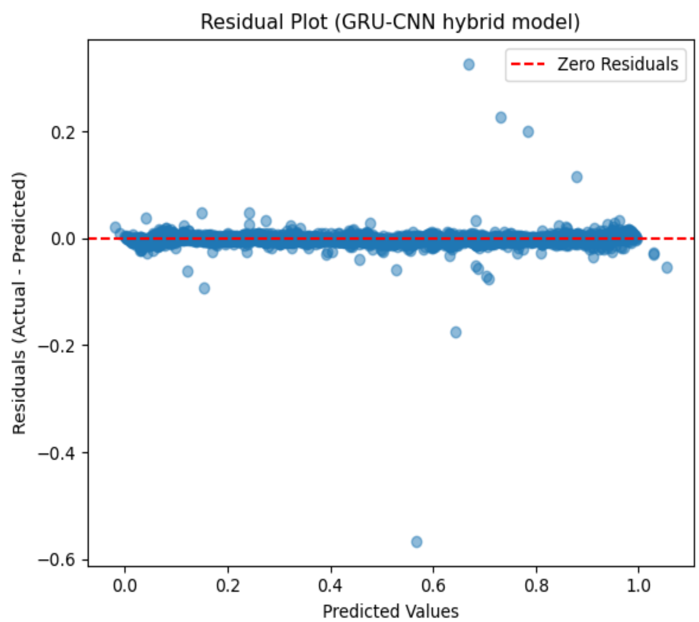
5.4.5 GRU-CNN Model

Various optimizers and learning rates were used to test the GRU-CNN hybrid model, which led to a range of final validation losses.

Table 8: 2-Layer GRU-CNN Training Model

Epochs	Batch size	Activation	Optimizer	Learning rate	Validation losses	Best Optimizer
50	128	relu and dense 1	Adam	0.001		Adam with Learning Rate 0.01 Best Validation Loss: 0.0007
			Adam	0.0001		
			RMSprop	0.001		
			RMSprop	0.0001		
			SGD	0.01		
			SGD	0.001		





```
Test Set Metrics (GRU-CNN hybrid model):
MSE: 0.0002
MAE: 0.0044
RMSE: 0.0141
MAPE: 5.2480
R2: 0.9977
Accuracy: 99.1106
Precision: 99.1106

Sample Predictions (10 points):
Actual vs Predicted Total_Paid:
Sample 1: Actual=0.08, Predicted=0.09
Sample 2: Actual=0.89, Predicted=0.88
Sample 3: Actual=0.36, Predicted=0.36
Sample 4: Actual=0.50, Predicted=0.50
Sample 5: Actual=0.10, Predicted=0.10
Sample 6: Actual=0.24, Predicted=0.24
Sample 7: Actual=0.59, Predicted=0.59
Sample 8: Actual=0.06, Predicted=0.06
Sample 9: Actual=0.66, Predicted=0.67
Sample 10: Actual=0.72, Predicted=0.72
```

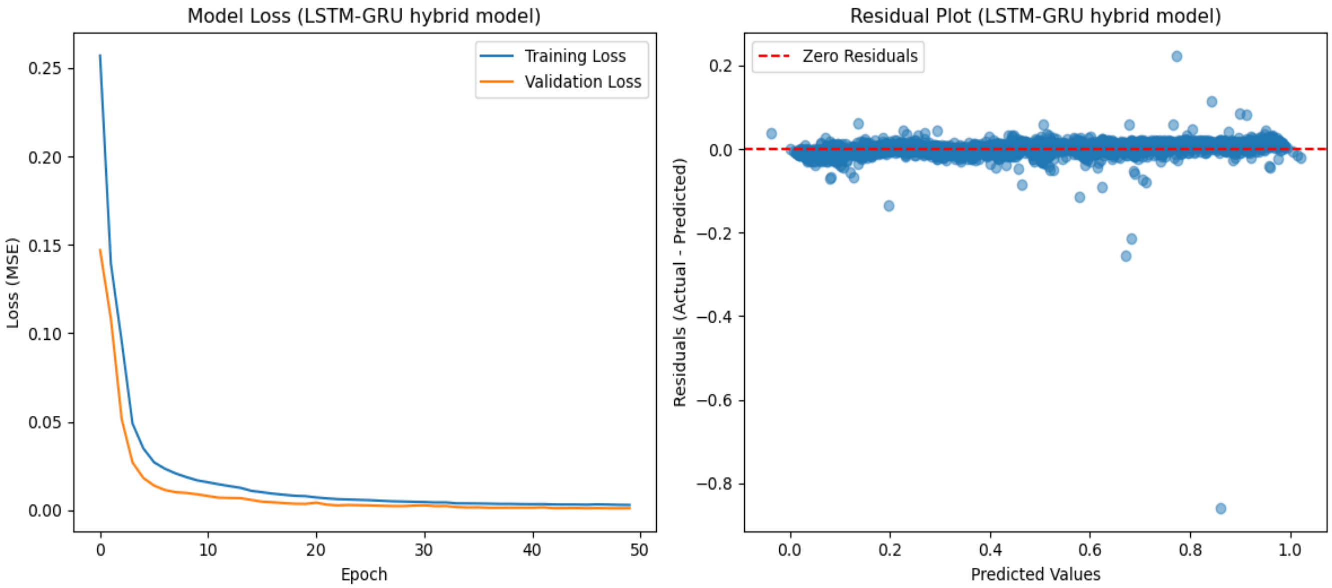
Figure 26: 2-Layer GRU-CNN Training Model output

5.4.6 LSTM-GRU Model

For the given objective, the hybrid LSTM-GRU model performs admirably, which led to a range of final validation losses. The sample predictions further corroborate the model's effectiveness, showing very close alignment between actual and predicted values.

Table 9: 2-Layer hybrid LSTM-GRU Training model

Epochs	Batch size	Activation	Optimizer	Learning rate	Validation losses	Best Optimizer
50	128	relu and dense 1	Adam	0.001	0.0023	Adam with Learning Rate: 0.0005 Best Validation Loss: 0.0011
			Adam	0.0001	0.0025	
			RMSprop	0.001	0.0026	
			RMSprop	0.0001	0.0025	
			SGD	0.01	0.0032	
			SGD	0.001	0.0731	




```
Test Set Metrics (LSTM-GRU hybrid model):
MSE: 0.0004
MAE: 0.0095
RMSE: 0.0196
MAPE: 5.8935
R2: 0.9956
Accuracy: 98.0950
Precision: 98.0950

Sample Predictions (10 points):
Actual vs Predicted Total_Paid:
Sample 1: Actual=0.08, Predicted=0.09
Sample 2: Actual=0.89, Predicted=0.88
Sample 3: Actual=0.36, Predicted=0.36
Sample 4: Actual=0.50, Predicted=0.49
Sample 5: Actual=0.10, Predicted=0.10
Sample 6: Actual=0.24, Predicted=0.24
Sample 7: Actual=0.59, Predicted=0.58
Sample 8: Actual=0.06, Predicted=0.07
Sample 9: Actual=0.66, Predicted=0.66
Sample 10: Actual=0.72, Predicted=0.72
```

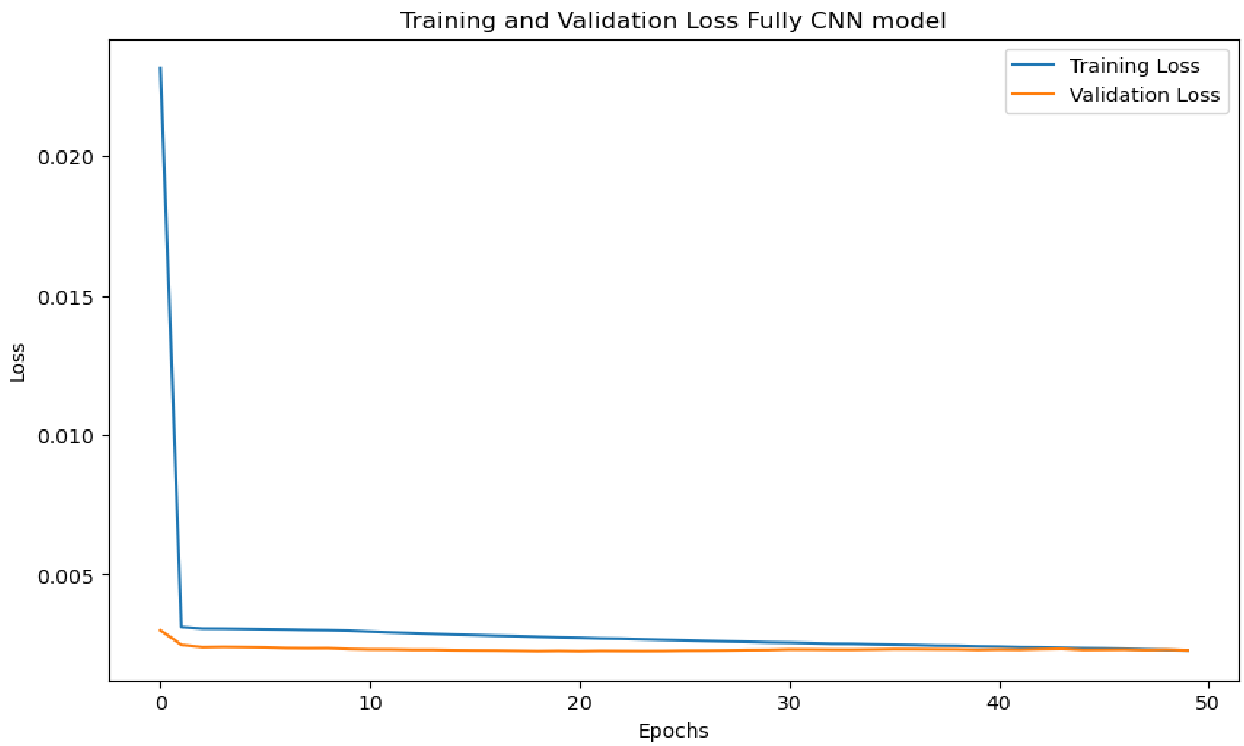
Figure 27: 2-Layer Hybrid of LSTM-GRU model Training output

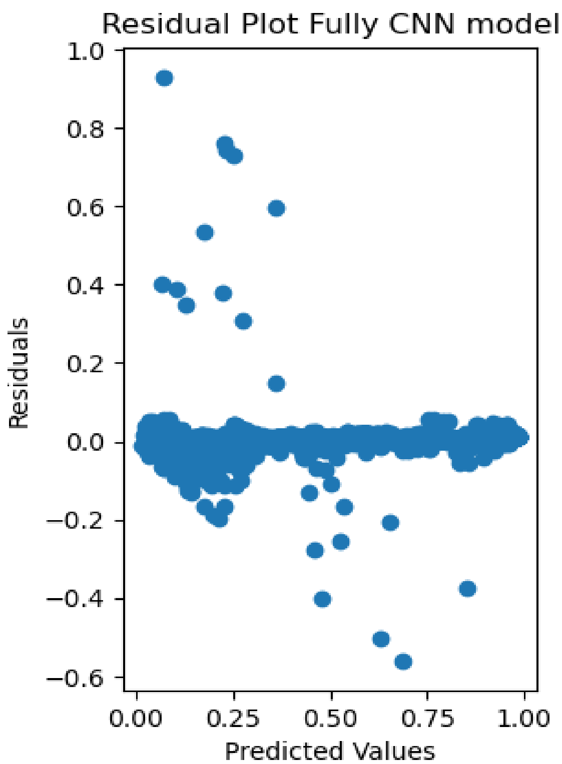
5.4.7 Fully CNN model

The fully convolutional neural network (Fully CNN) was evaluated using various optimizers and learning rates, resulting in different final validation losses.

Table 10: 2-Layer Fully CNN Training Model

Epochs	Batch size	Activation	Optimizer	Learning rate	Validation losses	Best Optimizer
50	128	relu and dense 1	Adam	0.001	0.0021	Adam with Learning Rate 0.001 Best Validation Loss: 0.0021
			Adam	0.0001	0.0023	
			RMSprop	0.001	0.0024	
			RMSprop	0.0001	0.0024	
			SGD	0.01	0.0026	
			SGD	0.001	0.0034	






```
Test Set Metrics (Best Model):
MSE: 0.0019
MAE: 0.0149
RMSE: 0.0442
MAPE: 23.2766
R2: 0.9779
Accuracy: 97.0217
Precision: 97.0217

Sample Predictions (10 points):
Actual vs Predicted Total_Paid:
Sample 1: Actual=0.22, Predicted=0.23
Sample 2: Actual=0.22, Predicted=0.24
Sample 3: Actual=0.89, Predicted=0.89
Sample 4: Actual=0.92, Predicted=0.92
Sample 5: Actual=0.50, Predicted=0.51
Sample 6: Actual=0.96, Predicted=0.97
Sample 7: Actual=0.79, Predicted=0.79
Sample 8: Actual=0.81, Predicted=0.81
Sample 9: Actual=0.23, Predicted=0.24
Sample 10: Actual=0.47, Predicted=0.47
```

Figure 28: 2-Layer Fully CNN Training Model output

5.4.8 FFNN model

The feedforward neural network (FFNN) was evaluated using various optimizers and learning rates, resulting in different final validation losses.

Table 11: 2-Layer FFNN Training Model

Epochs	**Batch size**	**Activation**	**Optimizer**	**Learning rate**	**Validation losses**	**Best Optimizer**
50	128	relu and dense 1	Adam	0.001	0.0024	RMSprop with Learning Rate 0.001 Best Validation Loss: 0.0024
			Adam	0.0001	0.0025	
			RMSprop	0.001	0.0024	
			RMSprop	0.0001	0.0026	
			SGD	0.01	0.0028	
			SGD	0.001	0.0048	

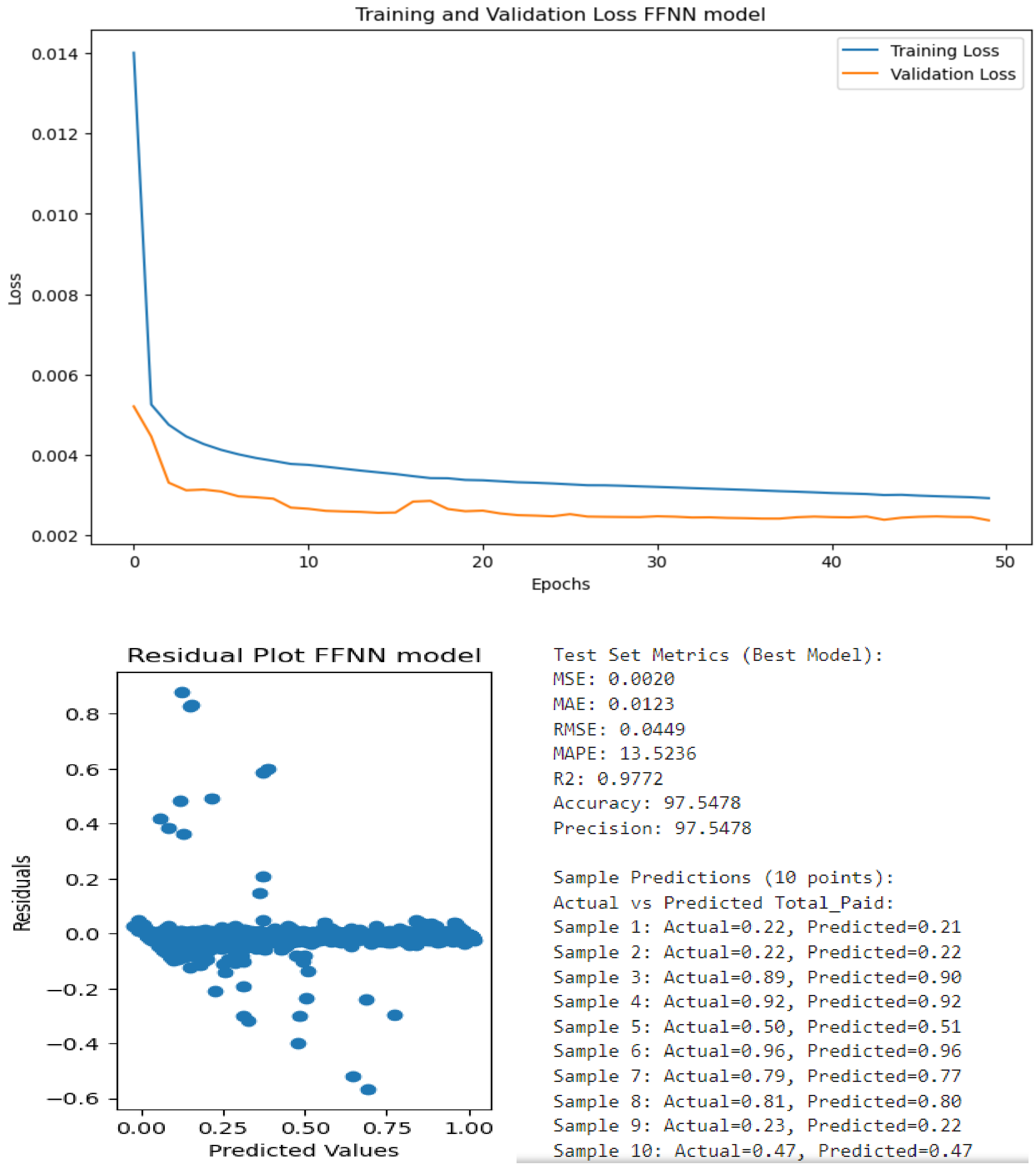


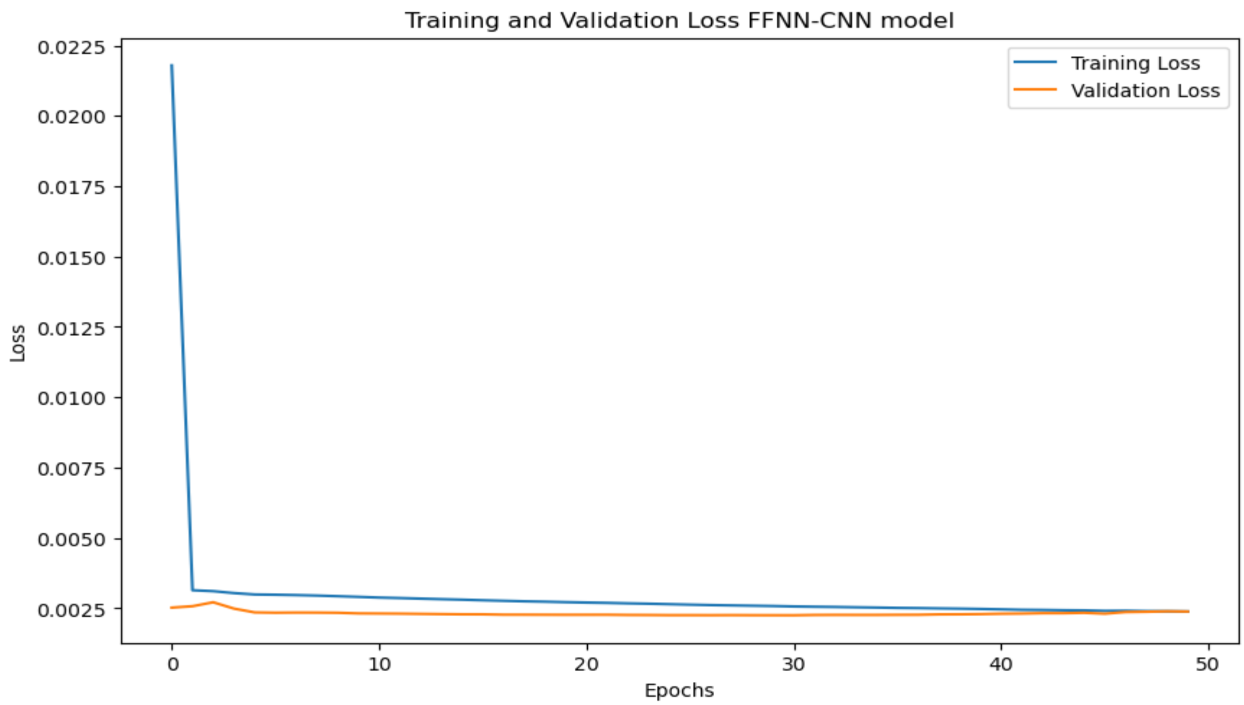
Figure 29: 2-Layer FFNN Training Model output

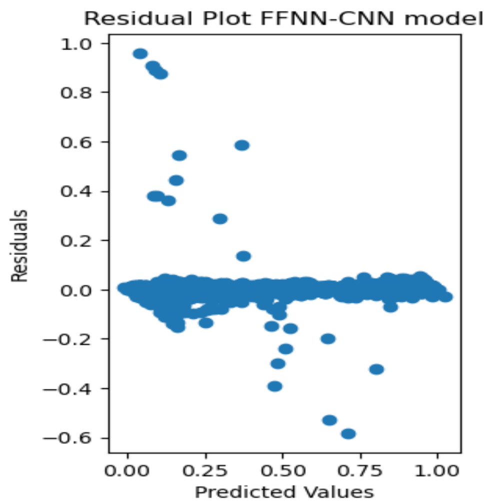
5.4.9 FFNN-CNN hybrid model

The FFNN-CNN hybrid model was evaluated across various optimizers and learning rates, yielding different final validation losses.

Table 12: 2-Layer FFNN-CNN Training model

Epochs	Batch size	Activation	Optimizer	Learning rate	Validation losses	Best Optimizer
50	128	relu and dense 1	Adam	0.001	0.0024	Adam with Learning Rate 0.001 Best Validation Loss: 0.0024
			Adam	0.0001	0.0024	
			RMSprop	0.001	0.0024	
			RMSprop	0.0001	0.0024	
			SGD	0.01	0.0025	
			SGD	0.001	0.0033	






```
Test Set Metrics (Best Model):
MSE: 0.0019
MAE: 0.0164
RMSE: 0.0442
MAPE: 24.4718
R2: 0.9779
Accuracy: 96.7297
Precision: 96.7297

Sample Predictions (10 points):
Actual vs Predicted Total_Paid:
Sample 1: Actual=0.22, Predicted=0.24
Sample 2: Actual=0.22, Predicted=0.26
Sample 3: Actual=0.89, Predicted=0.91
Sample 4: Actual=0.92, Predicted=0.93
Sample 5: Actual=0.50, Predicted=0.50
Sample 6: Actual=0.96, Predicted=0.98
Sample 7: Actual=0.79, Predicted=0.80
Sample 8: Actual=0.81, Predicted=0.82
Sample 9: Actual=0.23, Predicted=0.24
Sample 10: Actual=0.47, Predicted=0.48
```

Figure 30: 2-Layer FFNN-CNN Training Model output

5.5 Discussion

This investigation evaluates a variety of deep learning models, such as CNN, Fully CNN, LSTM-CNN, LSTM, GRU, GRU-CNN, FFNN-CNN, and FFNN, in order to predict Total Paid amounts in insurance claims. The results showed that different optimizers and learning rates performed differently.

Table 13: Model analysis evaluation

Model	Best Optimizer	Learning Rate	Validation Loss	Test MSE	Test MAE	Test RMSE	Test MAPE	Test R2	Test Accuracy	Test Precision
LSTM-CNN	Adam	0.001	0.0004	0.0002	0.0042	0.0142	4.9097	0.9977	99.1767	99.1767
CNN	RMSprop	0.001	0.0022	0.0019	0.0160	0.0441	21.0991	0.9780	96.8161	96.8161
Fully CNN	Adam	0.001	0.0023	0.0019	0.0149	0.0442	23.2766	0.9779	97.0217	97.0217
LSTM	RMSprop	0.0001	0.0007	0.0002	0.0058	0.0157	5.3168	0.9972	98.8399	98.8399
GRU	Adam	0.0001	0.0024	0.0021	0.0132	0.0453	17.4852	0.9767	97.3750	97.3750
LSTM-GRU	Adam	0.0005	0.0011	0.0004	0.0095	0.0196	5.8935	0.9956	98.0950	98.0950
GRU-CNN	Adam	0.01	0.0007	0.0002	0.0044	0.0141	5.2480	0.9977	99.1106	99.1106
FFNN-CNN	Adam	0.001	0.0024	0.0019	0.0164	0.0442	24.4718	0.9779	96.7297	96.7297
FFNN	RMSprop	0.001	0.0024	0.0020	0.0123	0.0449	13.5236	0.9772	97.5478	97.5478

The performance comparison of various deep learning models for a predictive task reveals that the LSTM-CNN Model, optimized with Adam at a learning rate of 0.001, outperforms others across multiple metrics, achieving the lowest validation loss (0.0004), MSE (0.0002), MAE (0.0041), RMSE (0.0142), and MAPE (4.9097), alongside the highest R^2 (0.9977), accuracy (99.1767), and precision (99.1767). The GRU-CNN model, also using Adam with a 0.001 learning rate, follows closely with strong performance (e.g., accuracy/precision of 99.1106 and MAPE of 5.2480). Other models, such as LSTM, CNN, and Fully CNN, show competitive but slightly inferior results, with LSTM (RMSprop, 0.0001) performing notably well (accuracy/ precision of 98.8399). Models like GRU and FFNN-CNN exhibit higher errors (e.g., test MAPE of 17.4852 and 24.4718, respectively) and lower accuracy, indicating that hybrid models, particularly LSTM-CNN and GRU-CNN, are more effective for this task, with Adam generally being the preferred optimizer.

5.6 Research Question Discussion

RQ1. Which variables is most significant in predicting insurance claims?

Out of 23 variables the most significant in predicting insurance claims is Total paid, the rest variables are used as input.

RQ2. What strategies can effectively improve the predictive power of deep learning models for insurance claims?

By hybrid model, Learning rate, and Optimizer/Hyperparameter.

RQ3. Which deep learning models can improve insurance claim prediction?

The LSTM-CNN and GRU-CNN Models emerged as the most effective

CHAPTER SEVEN
CONCLUSION AND FUTURE WORK

6.1 Conclusion

Six deep learning models (LSTM-CNN, CNN, LSTM, GRU, GRU-CNN, FFNN-CNN, and FFNN) have been thoroughly evaluated for the purpose of predicting insurance claims, and the results show that they have a great deal of promise. Strong predictive skills were continuously shown by all models, as shown by exceptionally high R-squared (R2) values and very low Mean Squared Error (MSE) on the test dataset. This demonstrates how well they explain the variation in actual insurance claims. Their practical utility is further reinforced by their great accuracy and precision across the board.

A critical observation was the extremely high Mean Absolute Percentage Error (MAPE) for all models. This metric's sensitivity to small actual values makes it less reliable for datasets where such values are common, despite good performance in absolute error metrics like MSE and MAE. Therefore, for insurance claim prediction, MSE, MAE, RMSE, and R-squared are more appropriate and informative performance indicators.

Among the evaluated models, the LSTM-CNN hybrid model emerged as the top performer, achieving the lowest validation losses, suggesting potentially better generalization. Overall, deep learning offers a robust approach to insurance claims prediction, providing highly accurate forecasts.

6.2 Contribution of this Study

Addressing knowledge gaps in current research and strengthening the shortcomings of earlier work are the goals of this study. The development of deep learning algorithm for insurance claims, problems with selection methods in standard machine learning, manipulation of dataset size and diversity, and application of the techniques for various insurance claim types. We apply methods via collecting diverse large dataset from Oromia Insurance S.C claims data and expanding it through up sampling method, implementing of deep learning model and improve overall insurance claim prediction. This study is among the first to apply hybrid DL models to Ethiopian insurance claims. In general, the contribution of this study is training the collected dataset, apply multiple deep learning model, and implementation of hyperparameter to increase performance.

6.3 Future Works

In light of the present results, future research can investigate a number of directions to improve the performance and comprehension of deep learning models for insurance claims prediction: Look more thoroughly than just learning rates and optimizers for the best hyperparameters, such as different hidden layer sizes, dropout rates, and batch sizes. The models may be able to learn more generalized features and close the gap between training and validation loss if they are trained on bigger and more varied datasets. Develop or adopt strategies to interpret these deep learning models interpreting a model's prediction is important for trust and follows laws in the insurance industry.

Future work can extend upon our present findings by focusing on these areas in order to design a more practical deep learning models for insurance claims prediction.

REFERENCE

Abdulkadir, U. I., & Fernando, A. (2024a). A Deep Learning Model for Insurance Claims Predictions. *Journal on Artificial Intelligence*, *6*(1), 71–83. https://doi.org/10.32604/jai.2024.045332

Abdulkadir, U. I., & Fernando, A. (2024b). A Deep Learning Model for Insurance Claims Predictions. *Journal on Artificial Intelligence*, *6*(1), 71–83. https://doi.org/10.32604/jai.2024.045332

Abdulkadir, U. I., & Fernando, A. (2024c). A Deep Learning Model for Insurance Claims Predictions. *Journal on Artificial Intelligence*, *6*(1), 71–83. https://doi.org/10.32604/jai.2024.045332

Abdulwahaab, N. (2024). Application of Explainable AI in Forgery Certificate. *Application of Explainable AI in Forgery Certificate*. https://doi.org/10.1007/S11042-02213797-W

Accurate insurance claims prediction with Deep Learning | by Lotus Labs | Medium. (n.d.). Retrieved May 6, 2025, from https://lotuslabs.medium.com/accurate-insurance-claims-prediction-with-deep-learning-23c575b90dd

Akrani, G. (2011). *Principles of Insurance - 7 Basic General Insurance Principles*. https://kalyan-city.blogspot.com/2011/03/principles-of-insurance-7-basic-general.html

Al., sharma et. (2022). *sharma et al. (2022) enhancing insurance claim prediction... - Google Scholar*. https://scholar.google.com/scholar?q=sharma+et+al.+(2022)+enhancing+insurance+claim+prediction+with+deep+learning-based+mlp+models&hl=en&as_sdt=0&as_vis=1&oi=scholart

Alamir, E., Urgessa, T., Hunegnaw, A., & Gopikrishna, T. (2021). *Motor Insurance Claim Status Prediction using Machine Learning Techniques*. *December*. https://doi.org/10.14569/IJACSA.2021.0120354

Alomar, K., Aysel, H. I., & Cai, X. (2024). *RNNs, CNNs and Transformers in Human Action Recognition: A Survey and a Hybrid Model*.

Amanda, C., & Pradipta, A. D. (2024). RISK-BASED PREMIUMS OF INSURANCE GUARANTEE SCHEMES: A MACHINE-LEARNING APPROACH. *Journal of Indonesian Economy and Business*, *39*(2), 121–142. https://doi.org/10.22146/jieb.v39i2.9323

Andrieş, A. M., & Căpraru, B. (2014). The nexus between competition and efficiency: The European banking industries experience. *International Business Review*, *23*(3), 566–579. https://doi.org/10.1016/J.IBUSREV.2013.09.004

Angima, C. B., & Mwangi, M. (2017). Effects of Underwriting and Claims Management on Performance of Property and Casualty Insurance Companies in East Africa. *European Scientific Journal, ESJ*, *13*(13), 358. https://doi.org/10.19044/esj.2017.v13n13p358

Asresa, T., Tigistu, G., & Bayih, M. (2023). *Large scale Multi-Labeled Aromatic Medcinal Plant Image classification using Deep Learnning*. https://doi.org/10.21203/RS.3.RS-3762562/V1

Awoke Advisor, G. (n.d.). *ADDIS ABABA UNIVERSITY COLLEGE OF BUSINESS AND ECONOMICS DEPARTMENT OF ACCOUNTING AND FINANCE Determinants of Life Insurance Demand in Ethiopia*.

Azize, T. (2015). *SCHOOL OF GRADUATE STUDIES CHALLENGES AND OPPORTUNITIES OF INSURANCE BUSINESS IN ETHIOPIA ADDIS ABEBA, ETHIOPIA*.

Baran, S., & Rola, P. (2022). Prediction of motor insurance claims occurrence as an imbalanced machine learning problem. *Papers*.

Bengio, Y., Lecun, Y., & Hinton, G. (2021). Deep learning for AI. *Communications of the ACM*, *64*(7), 58–65. https://doi.org/10.1145/3448250

Bhattacharya, S., Khanna, A., Ganapaneni, S., & Najana, M. (2024). Attention-Based Deep Learning Frameworks for Network Intrusion Detection: An Empirical Study. *International Journal of Global Innovations and Solutions (IJGIS)*. https://doi.org/10.21428/E90189C8.EB32676C

Birhanie, M. (2019). *ASSESSMENT OF MARKETING MIXES AND SALES PERFROMANCE OF INSURANCE COMPANIES: A CASE OF ETHIOPIAN INSURANCE CORPORATION THES IS SUBMITTED TO ADDIS ABABA UNIVERSITY COLLEGE OF BUSINESS AND ECONOMICS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF BUSINESS ADMINISTRATION (MBA)*.

Bulmer, M. (2024). *Multiple Regression*. The University of Queensland. https://uq.pressbooks.pub/portable-introduction-data-analysis/chapter/multiple-regression/

Burri, R. D., Burri, R., Bojja, R. R., & Buruga, S. R. (2019a). Insurance claim analysis using machine learning algorithms. *International Journal of Innovative Technology and Exploring Engineering*, *8*(6 Special Issue 4), 577–582. https://doi.org/10.35940/ijitee.F1118.0486S419

Burri, R. D., Burri, R., Bojja, R. R., & Buruga, S. R. (2019b). Insurance claim analysis using machine learning algorithms. *International Journal of Innovative Technology and Exploring Engineering*, *8*(6 Special Issue 4), 577–582. https://doi.org/10.35940/ijitee.F1118.0486S419

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247–1250. https://doi.org/10.5194/GMD-7-1247-2014

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, 1–24. https://doi.org/10.7717/PEERJ-CS.623/

Chollet, F. (2017). *Xception: Deep Learning With Depthwise Separable Convolutions* (pp. 1251–1258).

Chua, M., & Huber, K. E. (2023). Cahpter 27 - Regression. *Translational Radiation Oncology*, 145–152.

Conrad, E. (2011). Domain 5. *Eleventh Hour CISSP*, 69–87. https://doi.org/10.1016/B978-1-59749-566-0.00005-9

Cortis, D., Debattista, J., Debono, J., & Farrell, M. (2019a). InsurTech. *Palgrave Studies in Digital Business and Enabling Technologies*, 71–84. https://doi.org/10.1007/978-3-030-02330-0_5

Cortis, D., Debattista, J., Debono, J., & Farrell, M. (2019b). InsurTech. *Palgrave Studies in Digital Business and Enabling Technologies*, 71–84. https://doi.org/10.1007/978-3-030-02330-0_5

Customization of forecasting solutions. (2023).

Dhokane, Mr. R. (2024). CAR DAMAGE DETECTION USING CNN. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, *08*(04), 1–5. https://doi.org/10.55041/IJSREM30508

Dickson, G. R. ., Yarbro, C. Quinn., Lewitt, S. N. ., & Perry, Steve. (1991). *The harriers*. 258. https://www.goodreads.com/book/show/2037199.Of_War_and_Honor

Dinella, E., Mytkowicz, T., Svyatkovskiy, A., Bird, C., Naik, M., & Lahiri, S. (n.d.). *DeepMerge: Learning to Merge Programs*.

Dubey, A. K., & Jain, V. (2019a). Comparative Study of Convolution Neural Network's Relu and Leaky-Relu Activation Functions. *Lecture Notes in Electrical Engineering*, *553*, 873–880. https://doi.org/10.1007/978-981-13-6772-4_76

Dubey, A. K., & Jain, V. (2019b). Comparative Study of Convolution Neural Network's Relu and Leaky-Relu Activation Functions. *Lecture Notes in Electrical Engineering*, *553*, 873–880. https://doi.org/10.1007/978-981-13-6772-4_76

Enyinna Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*.

Favorskaya, M. N., & Andreev, V. V. (2019). The study of activation functions in deep learning for pedestrian detection and tracking. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2/W12), 53–59. https://doi.org/10.5194/ISPRS-ARCHIVES-XLII-2-W12-53-2019

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/J.PATREC.2005.10.010

Fjellstrom, C. (2022). Long Short-Term Memory Neural Network for Financial Time Series. *Proceedings - 2022 IEEE International Conference on Big Data, Big Data 2022*, 3496–3504. https://doi.org/10.1109/BigData55660.2022.10020784

Gamaleldin, W., Attayyib, O., Alnfiai, M. M., Alotaibi, F. A., & Ming, R. (2025). A hybrid model based on CNN-LSTM for assessing the risk of increasing claims in insurance companies. *PeerJ Computer Science*, *11*, 1–25. https://doi.org/10.7717/peerj-cs.2830

Ghasemi, Z., & Miandoab, P. S. (2024). Feasibility Study of Convolutional Long ShortTerm Memory Network for Pulmonary Movement Prediction in CT Images. *Journal of Biomedical Physics & Engineering*, *14*(1), 55. https://doi.org/10.31661/JBPE.V0I0.2105-1339

Gozuoglu, A., Ozgonenel, O., & Gezegin, C. (2024). CNN-LSTM based deep learning application on Jetson Nano: Estimating electrical energy consumption for future smart homes. *Internet of Things (Netherlands)*, *26*. https://doi.org/10.1016/j.iot.2024.101148

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Linear Methods for Regression*. 43–99. https://doi.org/10.1007/978-0-387-84858-7_3

Hnamte, V., & Hussain, J. (2023). Dependable intrusion detection system using deep convolutional neural network: A Novel framework and performance evaluation approach. *Telematics and Informatics Reports*, *11*, 100077. https://doi.org/10.1016/J.TELER.2023.100077

Ian Goodfellow, Yoshua Bengio, Aaron Courville, H. J. (2018). Deep learning. *Genetic Programming and Evolvable Machines*, *19*(1–2), 305–307.

Islam, M., Chen, G., & Jin, S. (2019). An Overview of Neural Network. *American Journal of Neural Networks and Applications*, *5*(1), 7. https://doi.org/10.11648/J.AJNNA.20190501.12

Kabeta, M. H. (2021). *ACCEPTANCE Application of Data Mining to Classify Medical Insurance Customers Based on Claim Experience: The Case of Awash Insurance Company S.C*.

Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, *64*(5), 402. https://doi.org/10.4097/KJAE.2013.64.5.402

Khinde, A., Kasture, P., Kandekar, D., Desai, V., & Zanje, M. (2023). *CAR DAMAGE DETECTION USING CNN ALGORITHM*. *11*(11), 2320–2882. www.ijcrt.org

Kim, B.-H., Sridharan, S., Atwal, A., & Ganapathi, V. (2020a). *Deep Claim: Payer Response Prediction from Claims Data with Deep Learning*.

Kim, B.-H., Sridharan, S., Atwal, A., & Ganapathi, V. (2020b). *Deep Claim: Payer Response Prediction from Claims Data with Deep Learning*. https://arxiv.org/abs/2007.06229v1

Kingma, D. P., & Ba, J. L. (2014a). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/pdf/1412.6980

Kingma, D. P., & Ba, J. L. (2014b). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Kiprotich Ng'elechei, J., Cheruiyot Chelule, J., Orango, H. I., & Anapapa, A. O. (2020). Modeling Frequency and Severity of Insurance Claims in an Insurance Portfolio. *American Journal of Applied Mathematics and Statistics*, *8*(3), 103–111. https://doi.org/10.12691/AJAMS-8-3-4

Krieger, F., Drews, P., & Velte, P. (2021). Explaining the (non-) adoption of advanced data analytics in auditing: A process theory. *International Journal of Accounting Information Systems*, *41*, 100511. https://doi.org/10.1016/j.accinf.2021.100511

Krishna Jampani, S. (2019). Fraud Detection in Insurance Claims Using AI. *Available Online Www.Jsaer.Com Journal of Scientific and Engineering Research 302 Journal of Scientific and Engineering Research*, *6*(1), 302–310.

Lecun, Y., Bengio, Y., & Hinton, G. (2015a). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. https://doi.org/10.1038/nature14539

Lecun, Y., Bengio, Y., & Hinton, G. (2015b). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/NATURE14539

Lecun, Y., Bengio, Y., & Hinton, G. (2015c). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/NATURE14539,

Lisovsky, A. L. (2021). APPLICATION OF NEURAL NETWORK TECHNOLOGIES FOR MANAGEMENT DEVELOPMENT OF SYSTEMS. *Strategic Decisions and Risk Management*, *11*(4), 378–389. https://doi.org/10.17747/2618-947X-923

Liu, M., Mroueh, Y., Ross, J., Zhang, W., Cui, X., Das, P., & Yang, T. (2019). Towards Better Understanding of Adaptive Gradient Algorithms in Generative Adversarial Nets. *8th International Conference on Learning Representations, ICLR 2020*.

Lu, Y. (2017). Deep neural networks and fraud detection. *U.U.D.M. Project Report*, 1–39.

Majnik, M., & Bosnić, Z. (2013). ROC analysis of classifiers in machine learning: A survey. *Intelligent Data Analysis*, *17*(3), 531–558. https://doi.org/10.3233/IDA-130592

Mary', S. T., & Gonfa, A. (2016). *PRACTICES AND CHALLENGES OF EMPLOYEE TRAINING AT COMMERCIAL BANK OF ETHIOPIA*. http://repository.smuc.edu.et/handle/123456789/3598

Mienye, I. D., & Jere, N. (2024a). Deep Learning for Credit Card Fraud Detection: A Review of Algorithms, Challenges, and Solutions. *IEEE Access*, *12*, 96893–96910. https://doi.org/10.1109/ACCESS.2024.3426955

Mienye, I. D., & Jere, N. (2024b). Deep Learning for Credit Card Fraud Detection: A Review of Algorithms, Challenges, and Solutions. *IEEE Access*, *12*, 96893–96910. https://doi.org/10.1109/ACCESS.2024.3426955

Mienye, I. D., Sun, Y., & Ileberi, E. (2024). Artificial intelligence and sustainable development in Africa: A comprehensive review. *Machine Learning with Applications*, *18*, 100591. https://doi.org/10.1016/J.MLWA.2024.100591

Mienye, I. D., & Swart, T. G. (2024a). A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications. *Information 2024, Vol. 15, Page 755*, *15*(12), 755. https://doi.org/10.3390/INFO15120755

Mienye, I. D., & Swart, T. G. (2024b). A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications. *Information 2024, Vol. 15, Page 755*, *15*(12), 755. https://doi.org/10.3390/INFO15120755

Mienye, I. D., & Swart, T. G. (2024c). A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications. *Information 2024, Vol. 15, Page 755*, *15*(12), 755. https://doi.org/10.3390/INFO15120755

Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information 2024, Vol. 15, Page 517*, *15*(9), 517. https://doi.org/10.3390/INFO15090517

Nam, B., Kim, J. Y., Kim, I. Y., & Cho, B. H. (2022). Selective Prediction With Long Short-term Memory Using Unit-Wise Batch Standardization for Time Series Health Data Sets: Algorithm Development and Validation. *JMIR Medical Informatics*, *10*(3). https://doi.org/10.2196/30587

Navatha, K., Chandan Babu, N., & Hara Gopal, V. V. (2023). "Forecasting Insurance Claims using Deep Learning approach." *International Journal of Current Science (IJCSPUB) Www.Ijcspub.Org*, *13*(1), 668. www.ijcspub.org

Nielsen, M. (n.d.). *Neural Networks and Deep Learning*. Retrieved September 23, 2025, from www.liuhao.me

O'Halloran, T., Obaido, G., Otegbade, B., & Mienye, I. D. (2024). A deep learning approach for Maize Lethal Necrosis and Maize Streak Virus disease detection. *Machine Learning with Applications*, *16*, 100556. https://doi.org/10.1016/J.MLWA.2024.100556

Oromia Insurance - Rest. Assured. (n.d.-a). Retrieved March 29, 2025, from https://oromiainsurance.com/index.php

Oromia Insurance - Rest. Assured. (n.d.-b). Retrieved March 29, 2025, from https://oromiainsurance.com/index.php

Outreville, J. F. (1998). Insurance Concepts. *Theory and Practice of Insurance*, 131–146. https://doi.org/10.1007/978-1-4615-6187-3_8

Pfeffer. (1956). *Insurance and economic theory : Pfeffer, Irving : Free Download, Borrow, and Streaming : Internet Archive*. https://archive.org/details/insuranceeconomi00pfef

Polamuri, S. R., Srinivas, D. K., & Krishna Mohan, D. A. (2022). Multi-Model Generative Adversarial Network Hybrid Prediction Algorithm (MMGAN-HPA) for stock market prices prediction. *Journal of King Saud University - Computer and Information Sciences*, *34*(9), 7433–7444. https://doi.org/10.1016/J.JKSUCI.2021.07.001

Pum, M. (2025). *Deep Learning in Insurance Claim Automation for Car Accidents*. *December 2024*.

Rawat, S., Rawat, A., Kumar, D., & Sabitha, A. S. (2021). Application of machine learning and data visualization techniques for decision support in the insurance sector. *International Journal of Information Management Data Insights*, *1*(2). https://doi.org/10.1016/J.JJIMEI.2021.100012

Rejda, G. E., McNamara, M. J., & Rabel, W. H. (2021a). *Principles of Risk Management and Insurance, Global Editon 14th Edition*.

Rejda, G. E., McNamara, M. J., & Rabel, W. H. (2021b). *Principles of Risk Management and Insurance, Global Editon 14th Edition*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature 1986 323:6088*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Saputro, A. R., Murfi, H., & Nurrohmah, S. (2019). Analysis of Deep Neural Networks for Automobile Insurance Claim Prediction. *Communications in Computer and Information Science*, *1071*, 114–123. https://doi.org/10.1007/978-981-32-9563-6_12

Saripalli, P., Tirumala, V., & Chimmad, A. (2017). Assessment of healthcare claims rejection risk using machine learning. *2017 IEEE 19th International Conference on E-Health Networking, Applications and Services, Healthcom 2017*, *2017-December*, 1–6. https://doi.org/10.1109/HEALTHCOM.2017.8210758

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/J.NEUNET.2014.09.003

Schmidt, R. M. (n.d.). *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*.

Schneider, J., & Vlachos, M. (n.d.). *Personalization of Deep Learning*.

Sharma, D. K., Chatterjee, M., Kaur, G., & Vavilala, S. (2022). Deep learning applications for disease diagnosis. *Deep Learning for Medical Applications with Unique Data*, 31–51. https://doi.org/10.1016/B978-0-12-824145-5.00005-8

Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, *4*, 310–316.

Sherstinsky, A. (n.d.). *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*. Retrieved September 23, 2025, from https://www.linkedin.com/in/alexsherstinsky

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/J.PHYSD.2019.132306

Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, *7*, 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

Singh, M., & Sharma, S. (2020). (2020). *Singh, M., & Sharma, S. (2020). Critical Success Factors in Construction Projects. International Journal of Recent Technology and Engineering, 8, 1799-1804. - References - Scientific Research Publishing*. https://www.scirp.org/reference/referencespapers?referenceid=3663570

Soltani, R., & Jiang, H. (2016). *Higher Order Recurrent Neural Networks*.

SPENCE, M., & ZECKHAUSER, R. (1978). Insurance, Information, and Individual Action. *Uncertainty in Economics*, 333–343. https://doi.org/10.1016/B978-0-12-214850-7.50027-9

St. Mary's University Institutional Repository: Hailu Zeleke 2007. Insurance in Ethiopia: Historical Development, Present Status and Future Challenges. Addis Ababa: Master Printing Press. (n.d.). Retrieved March 29, 2025, from http://repository.smuc.edu.et/handle/123456789/2712

Surya, H. A., Sukono, Napitupulu, H., & Ismail, N. (2024). A Systematic Literature Review of Insurance Claims Risk Measurement Using the Hidden Markov Model. *Risks*, *12*(11), 1–18. https://ideas.repec.org/a/gam/jrisks/v12y2024i11p169-d1504642.html

Sven Groen. (2023a). *Erasmus University Rotterdam Erasmus School of Economics Master Thesis Data Science & Marketing Analytics Predicting insurance premiums with Machine Learning : the MS Amlin case Name : Sven Groen Student number : 478036 Supervisor : C . Cavicchia Second A*.

Sven Groen. (2023b). *Erasmus University Rotterdam Erasmus School of Economics Master Thesis Data Science & Marketing Analytics Predicting insurance premiums with Machine Learning : the MS Amlin case Name : Sven Groen Student number : 478036 Supervisor : C . Cavicchia Second A*.

Talesh, S. A. (2015). Insurance and the Law. *International Encyclopedia of the Social & Behavioral Sciences: Second Edition*, 215–220. https://doi.org/10.1016/B978-0-08-097086-8.86037-7

van Engelen, J. E., Hoos, H. H., Fawcett Jesper E van Engelen, T. B., & Hoos hh, H. H. (2020). A survey on semi-supervised learning. *Springer*, *109*(2), 373–440. https://doi.org/10.1007/s10994-019-05855-6

Wang, S., Tang, J., & Liu, H. (2016). Feature Selection. *Encyclopedia of Machine Learning and Data Mining*, 1–9. https://doi.org/10.1007/978-1-4899-7502-7_101-1

Wang, W. L., Moore, J. K., Martiny, A. C., & Primeau, F. W. (2019). Convergent estimates of marine nitrogen fixation. *Nature*, *566*(7743), 205–211. https://doi.org/10.1038/S41586-019-0911-2

Watt, A., David Wiley, et al., Resources, P. M. O., & TAP-a-PM. (2014). *16. Risk Management Planning*. BCcampus.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). Deep learning. *Data Mining*, 417–466. https://doi.org/10.1016/B978-0-12-804291-5.00010-6

Yusuf, T. O., Ajemunigbohun, S. S., Alli, G. N., Yusuf, T. O., Ajemunigbohun, S. S., & Alli, G. N. (2017). A Critical Review of Insurance Claims Management: A Study of Selected Insurance Companies in Nigeria. *SPOUDAI Journal of Economics and Business*, *67*(2), 69–84. https://econpapers.repec.org/RePEc:spd:journl:v:67:y:2017:i:2:p:69-84

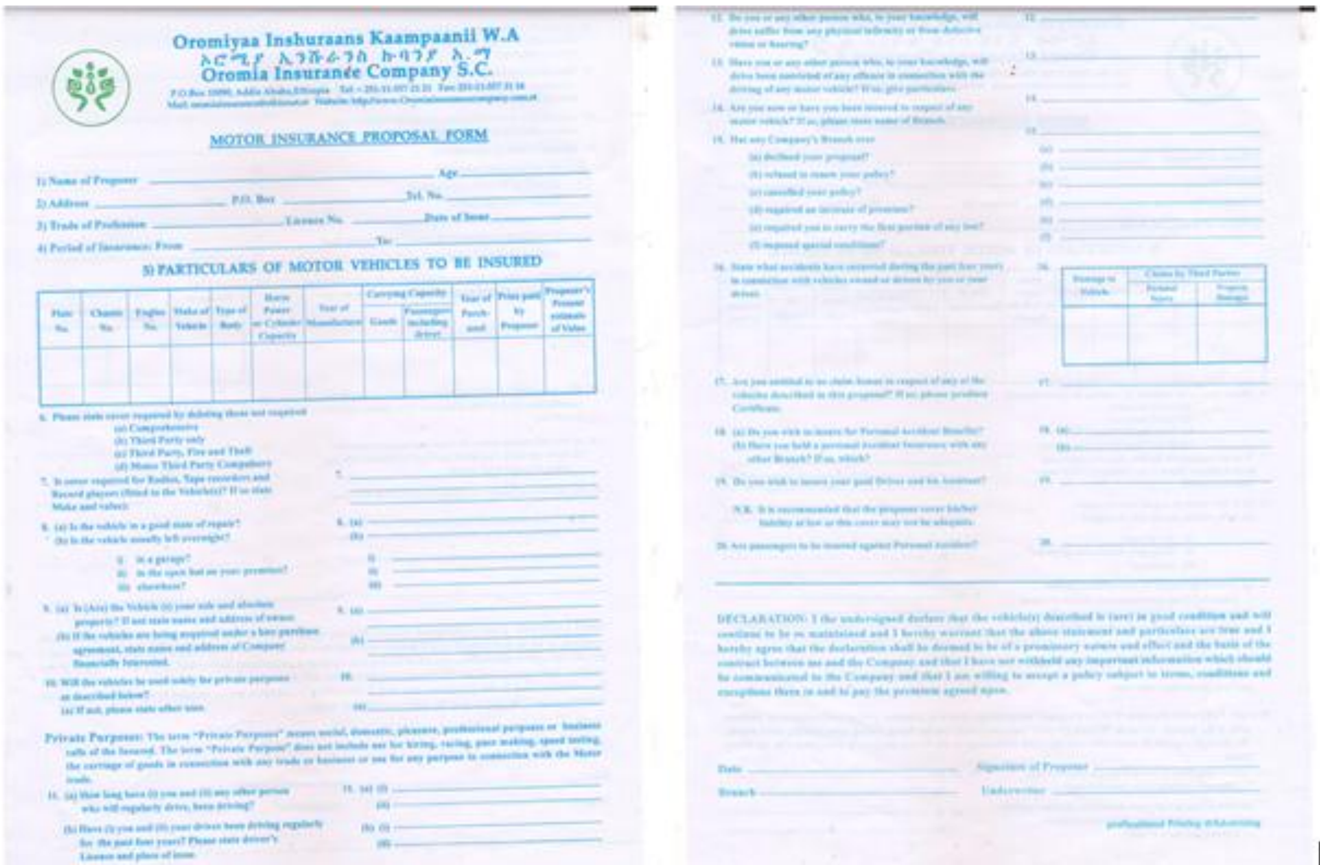
Zeleke, H. (2007). *SCHOOL OF GRADUATE STUDIES CHALLENGES AND OPPORTUNITIES OF INSURANCE BUSINESS IN ETHIOPIA ADDIS ABEBA, ETHIOPIA*.

Zhao, Y., Shen, Y., & Yao, J. (2019). Recurrent neural network for text classification with hierarchical multiscale dense connections. *IJCAI International Joint Conference on Artificial Intelligence*, *2019-August*, 5450–5456. https://doi.org/10.24963/IJCAI.2019/757

Zhu, X., & Zhang, F. (2016). *神经网络与深度学习 Neural Networks and Deep Learning （美）Michael Nielsen 著*.

APPENDIX

*Appendic 1: Underwriting agreement form(*Oromia Insurance - Rest. Assured.*, n.d.-a)*

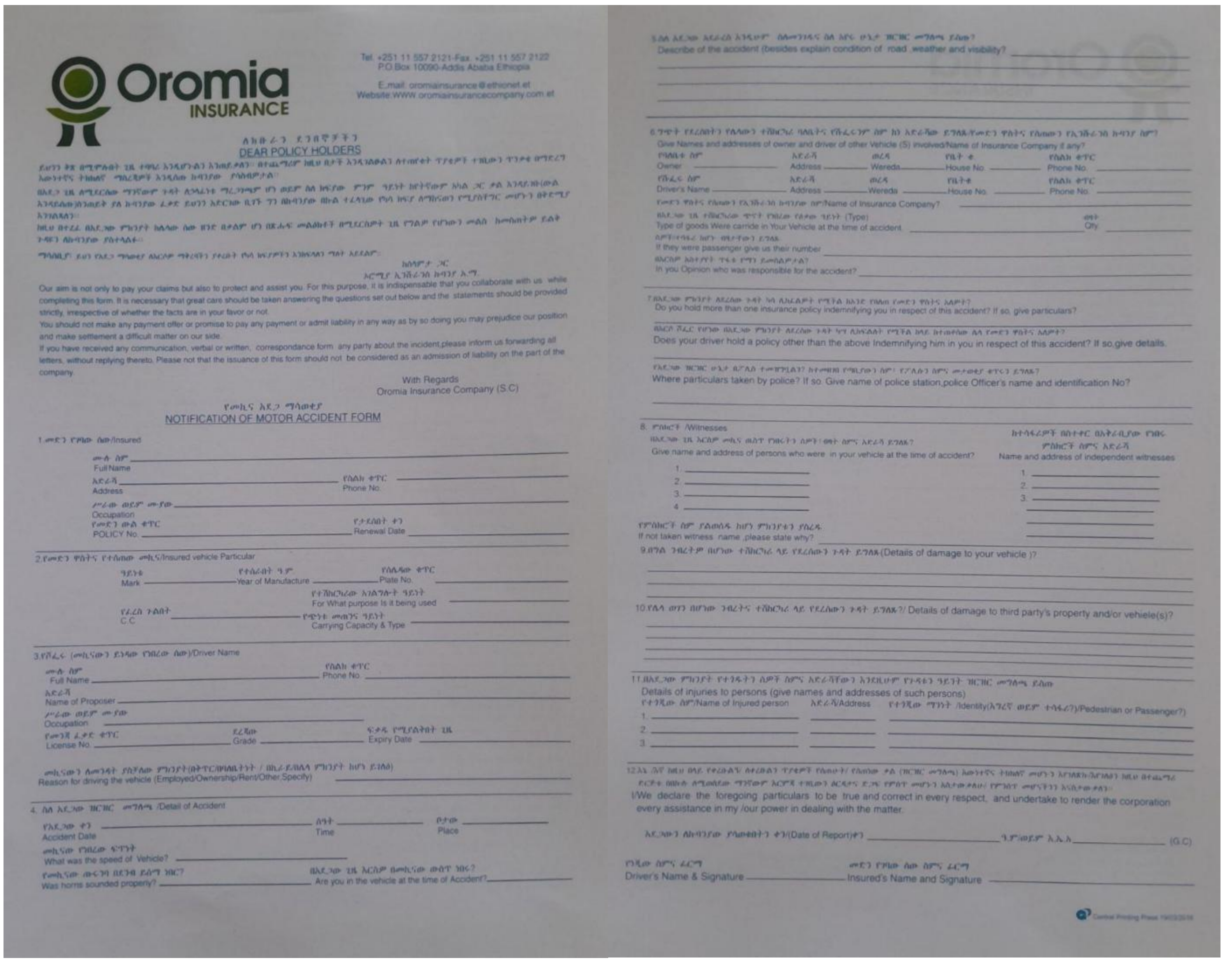

Oromiyaa Inshuraans Kaampaanii W.A

Oromia Insurance Company S.C.

MOTOR INSURANCE PROPOSAL FORM

5) PARTICULARS OF MOTOR VEHICLES TO BE INSURED

*Appendic 2: Claim report form. Source(*Oromia Insurance - Rest. Assured.*, n.d.-a)*



Oromia INSURANCE

Tel. +251 11 557 2121-Fax. +251 11 557 2122
P.O.Box 10090-Addis Ababa Ethiopia
E.mail: oromiainsurance@ethionet.et
Website:WWW.oromiainsurancecompany.com.et

DEAR POLICY HOLDERS

Our aim is not only to pay your claims but also to protect and assist you. For this purpose, it is indispensable that you collaborate with us while completing this form. It is necessary that great care should be taken answering the questions set out below and the statements should be provided strictly, irrespective of whether the facts are in your favor or not.
You should not make any payment offer or promise to pay any payment or admit liability in any way as by so doing you may prejudice our position and make settlement a difficult matter on our side.
If you have received any communication, verbal or written, correspondance form any party about the incident,please inform us forwarding all letters, without replying thereto. Please not that the issuance of this form should not be considered as an admission of liability on the part of the company.

With Regards
Oromia Insurance Company (S.C)

NOTIFICATION OF MOTOR ACCIDENT FORM

1. Insured
Full Name
Address
Phone No.
Occupation
POLICY No.
Renewal Date

2. Insured vehicle Particular
Mark
Year of Manufacture
Plate No.
For What purpose Is it being used
C.C
Carrying Capacity & Type

3. Driver Name
Full Name
Phone No.
Name of Proposer
Occupation
License No.
Grade
Expiry Date
Reason for driving the vehicle (Employed/Ownership/Rent/Other,Specify)

4. Detail of Accident
Accident Date
Time
Place
What was the speed of Vehicle?
Was horns sounded properly?
Are you in the vehicle at the time of Accident?

5. Describe of the accident (besides explain condition of road weather and visibility?

6. Give Names and addresses of owner and driver of other Vehicle (S) involved/Name of Insurance Company if any?
Owner
Address
Wereda
House No.
Phone No.
Driver's Name
Address
Wereda
House No.
Phone No.
Name of Insurance Company?
Type of goods Were carride in Your Vehicle at the time of accident (Type)
Qty
If they were passenger give us their number
In you Opinion who was responsible for the accident?

7. Do you hold more than one insurance policy indemnifying you in respect of this accident? If so, give particulars?
Does your driver hold a policy other than the above Indemnifying him in you in respect of this accident? If so,give details.
Where particulars taken by police? If so Give name of police station,police Officer's name and identification No?

8. Witnesses
Give name and address of persons who were in your vehicle at the time of accident?
1.
2.
3.
4.
Name and address of independent witnesses
1.
2.
3.
If not taken witness name ,please state why?

9. (Details of damage to your vehicle)?

10. Details of damage to third party's property and/or vehiele(s)?

11. Details of injuries to persons (give names and addresses of such persons)
Name of Injured person
Address
Identity(Pedestrian or Passenger?)
1.
2.
3.

12. I/We declare the foregoing particulars to be true and correct in every respect, and undertake to render the corporation every assistance in my /our power in dealing with the matter.

(Date of Report)
(G.C)

Driver's Name & Signature
Insured's Name and Signature

Appendic 3: Import library and Build LSTM-CNN hybrid model

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import RobustScaler, LabelEncoder
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Conv1D, Dense, Dropout, Flatten
from tensorflow.keras.optimizers import Adam, RMSprop, SGD, AdamW
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping
import matplotlib.pyplot as plt
from datetime import datetime
# Set random seed for reproducibility
tf.random.set_seed(42)
np.random.seed(42)
# Load and preprocess data
def preprocess_data(file_path):
try:
df = pd.read_csv(file_path)
except FileNotFoundError:
raise FileNotFoundError("Claim_data_final.csv not found. Please ensure the file is in the correct
directory.")
# Handle missing values
numerical_cols = ['Estimate_Payment', 'Estimate_Recovery', 'Total_Estimate',
```

```
'Paid_Payment', 'Paid_Recovery', 'Loss_date', 'Claim_no', 'Policy_no', 'Total_Paid']
categorical_cols = ['Branch', 'Class', 'Product', 'Insured_name', 'Closed']
df[numerical_cols] = df[numerical_cols].fillna(df[numerical_cols].mean())
df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().iloc[0])
94# Convert Loss_date to numerical format (days since epoch)
df['Loss_date'] = pd.to_datetime(df['Loss_date'])
df['Loss_date'] = (df['Loss_date'] - pd.Timestamp("1970-01-01")).dt.days
# Feature engineering
df['Payment_Estimate_Ratio'] = df['Paid_Payment'] / (df['Estimate_Payment'] + 1e-10)
df['Recovery_Estimate_Ratio'] = df['Paid_Recovery'] / (df['Estimate_Recovery'] + 1e-10)
df['Estimate_to_Total_Paid'] = df['Total_Estimate'] / (df['Total_Paid'] + 1e-10)
# Encode categorical variables
categorical_cols = ['Branch', 'Class', 'Product', 'Insured_name', 'Closed']
label_encoders = {}
for col in categorical_cols:
le = LabelEncoder()
df[col] = le.fit_transform(df[col].astype(str))
label_encoders[col] = le
# Features and target
features = ['Branch', 'Class', 'Product', 'Claim_no', 'Policy_no', 'Insured_name', 'Estimate_Payment',
'Estimate_Recovery', 'Total_Estimate', 'Paid_Payment', 'Paid_Recovery', 'Loss_date', 'Closed',
'Payment_Estimate_Ratio', 'Recovery_Estimate_Ratio', 'Estimate_to_Total_Paid']
X = df[features].values
y = np.log1p(df['Total_Paid'].values) # Log-transform target to handle skewness
# Scale features and target
scaler_X = RobustScaler()
X = scaler_X.fit_transform(X)
scaler_y = RobustScaler()
y = scaler_y.fit_transform(y.reshape(-1, 1)).flatten()
# Reshape for LSTM-CNN [samples, timesteps, features]
X = X.reshape((X.shape[0], X.shape[1], 1))
return X, y, scaler_y, df
```

```
# Split data
95def split_data(X, y):
# First split: 80% train+val, 10% test
X_temp, X_test, y_temp, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
# Second split: 88.89% train, 11.11% val (to get 80:10 from original)
X_train, X_val, y_train, y_val = train_test_split(X_temp, y_temp, test_size=0.1111, random_state=42)
print(f"Train set size: {len(X_train)}, Validation set size: {len(X_val)}, Test set size: {len(X_test)}")
return X_train, X_val, X_test, y_train, y_val, y_test
# Build LSTM-CNN hybrid model
def build_model(input_shape, optimizer):
model = Sequential([
LSTM(64, input_shape=input_shape, return_sequences=True),
Conv1D(filters=32, kernel_size=3, activation='relu', padding='same'),
Dropout(0.1),
Flatten(),
Dense(16, activation='relu'),
Dense(1)
])
model.compile(optimizer=optimizer, loss='mse')
return model
# Calculate metrics including custom Accuracy and Precision
def calculate_metrics(y_true, y_pred):
mse = mean_squared_error(y_true, y_pred)
mae = mean_absolute_error(y_true, y_pred)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((y_true - y_pred) / np.where(y_true == 0, 1, y_true))) * 100
r2 = r2_score(y_true, y_pred)
# Custom Accuracy and Precision: (1 - MAE / mean_y) * 100
mean_y = np.mean(y_true)
accuracy = (1 - mae / mean_y) * 100 if mean_y != 0 else 0
precision = accuracy
```

96return { 'MSE': mse, 'MAE': mae, 'RMSE': rmse, 'MAPE': mape, 'R2': r2, 'Accuracy': accuracy, 'Precision': precision,

}

Check for overfitting/underfitting

def check_fitting(history):

train_loss = history.history['loss'][-1]

val_loss = history.history['val_loss'][-1]

print("\nOverfitting/Underfitting Analysis:")

if val_loss > train_loss * 1.2:

print("Potential Overfitting: Validation loss is significantly higher than training loss.")

elif train_loss > val_loss * 1.2:

print("Potential Underfitting: Training loss is significantly higher than validation loss.")

else:

print("Good Fit: Training and validation losses are relatively close.")

Hyperparameter tuning with different optimizers and learning rates

def hyperparameter_tuning(X_train, X_val, y_train, y_val, input_shape):

optimizers = [

('Adam', Adam, [0.01, 0.005, 0.001, 0.0005]), ('RMSprop', RMSprop, [0.01, 0.005, 0.001, 0.0005]), ('SGD', SGD, [0.05, 0.01, 0.005])

]

best_model = None

best_val_loss = float('inf')

best_config = None

97best_history = None

Callbacks

lr_scheduler = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=5)

early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

for opt_name, opt_class, learning_rates in optimizers:

for lr in learning_rates:

print(f"\nTraining with {opt_name}, Learning Rate: {lr}")

optimizer = opt_class(learning_rate=lr, clipnorm=1.0)

model = build_model(input_shape, optimizer)

```
history = model.fit(X_train, y_train,
validation_data=(X_val, y_val),
epochs=50, batch_size=128, callbacks=[lr_scheduler, early_stopping], verbose=0)
for epoch in range(0, 50, 10):
if epoch < len(history.history['loss']):
print(f"Epoch {epoch + 1}/50, Train Loss: {history.history['loss'][epoch]:.4f}, Val Loss:
{history.history['val_loss'][epoch]:.4f}")
val_loss = history.history['val_loss'][-1]
print(f"Final Validation Loss: {val_loss:.4f}")
if val_loss < best_val_loss:
best_val_loss = val_loss
best_model = model
best_config = (opt_name, lr)
best_history = history
print(f"\nBest Configuration: {best_config[0]} with Learning Rate: {best_config[1]}")
print(f"Best Validation Loss: {best_val_loss:.4f}")
return best_model, best_history
98# Main execution
def main():
# Load and preprocess data
X, y, scaler_y, df = preprocess_data('Claim_data_final.csv')
# Split data
X_train, X_val, X_test, y_train, y_val, y_test = split_data(X, y)
# Hyperparameter tuning
best_model, history = hyperparameter_tuning(X_train, X_val, y_train, y_val, (X_train.shape[1],
X_train.shape[2]))
# Check for overfitting/underfitting
check_fitting(history)
# Make predictions with best model
y_pred_scaled = best_model.predict(X_test, verbose=0)
# Inverse transform predictions and actual values
y_pred = np.expm1(scaler_y.inverse_transform(y_pred_scaled.reshape(-1, 1)).flatten()) # Inverse log
```

```
transform
y_test_orig = np.expm1(scaler_y.inverse_transform(y_test.reshape(-1, 1)).flatten()) # Inverse log
transform
# Verify variable shapes
print(f"y_test_orig shape: {y_test_orig.shape}, y_pred shape: {y_pred.shape}")
# Calculate metrics
metrics = calculate_metrics(y_test_orig, y_pred)
# Print metrics
print("\nTest Set Metrics (LSTM-CNN hybrid model):")
for metric, value in metrics.items():
print(f"{metric}: {value:.4f}")
# Print 10 sample predictions
print("\nSample Predictions (10 points):")
print("Actual vs Predicted Total_Paid:")
for i in range(min(10, len(y_test_orig))):
print(f"Sample {i+1}: Actual={y_test_orig[i]:.2f}, Predicted={y_pred[i]:.2f}")
99# Plot training history and residual plot
plt.figure(figsize=(12, 5))
# Subplot 1: Training and Validation Loss
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss (LSTM-CNN hybrid model)')
plt.xlabel('Epoch')
plt.ylabel('Loss (MSE)')
plt.legend()
# Subplot 2: Residual Plot
plt.subplot(1, 2, 2)
residuals = y_test_orig - y_pred
plt.scatter(y_pred, residuals, alpha=0.5)
plt.axhline(0, color='red', linestyle='--', label='Zero Residuals')
plt.title('Residual Plot (LSTM-CNN hybrid model)')
```

```
plt.xlabel('Predicted Values')
plt.ylabel('Residuals (Actual - Predicted)')
plt.legend()
plt.tight_layout()
plt.savefig('lstm_cnn_model_plots.png')
plt.show()
if __name__ == "__main__":
main()
```