Incremental Feature Selection for High-Dimensional Data Streams

## A PROJECT REPORT

***Submitted by***

**M Likhith Reddy**

**(Reg. No. CH.SC.U4AIE23033)**

**N Harshith Varma**

**(Reg. No. CH.SC.U4AIE23035)**

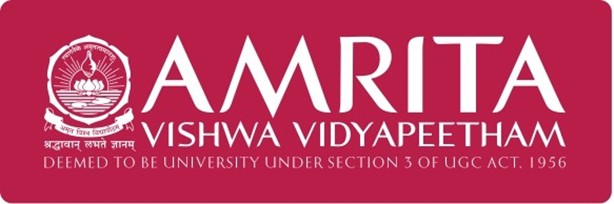
**N Charan**

**(Reg. No. CH.SC.U4AIE23038)**

***In partial fulfillment for the award of the degree of***

## BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

***Under the guidance of* Dr BharathiMohan Submitted to**

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AMRITA SCHOOL OF COMPUTING**

**AMRITA VISHWA VIDYAPEETHAM CHENNAI - 601103**

**APRIL 2025**



**BONAFIDE CERTIFICATE**

This is to certify that this project report entitled **“Incremental Feature Selection for High- Dimensional Data Streams”** is the bonafide work of **Mr. M Likhith Reddy (Reg. No. CH.SC.U4AIE23033), Mr. N Harshith Varma (Reg. No. CH.SC.U4AIE23035), Mr. N**

**Charan (Reg. No. CH.SC.U4AIE23038)** who carried out the project work under my supervi- sion as a part of the End Semester Project for the course 22AIE213 - Machine Learning.

## SIGNATURE

**Dr BharathiMohan**

**Assistant Professor (Sr.Gr.)**

Department of Computer Science and Engineering Amrita School of Computing,

Amrita Vishwa Vidyapeetham, Chennai Campus

**Name Signature**

M Likhith Reddy (Reg.No.CH.SC.U4AIE23033)

N Harshith Varma (Reg.No.CH.SC.U4AIE23035)

N Charan (Reg.No.CH.SC.U4AIE23038)



## DECLARATION BY THE CANDIDATE

I declare that the report entitled **“Incremental Feature Selection for High-Dimensional Data Streams”** submitted by me for the degree of Bachelor of Technology is the record of the project work carried out by me as a part of End semester project for the course 22AIE213 - Machine Learning under the guidance of **“Dr Bharathi Mohan”** and this work has not formed the basis for the award of any course project, degree, diploma, associateship, fellowship, titled in this or any other University or other similar institution of higher learning. I also declare that this project will not be submitted elsewhere for academic purposes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Register Number** | **Name** | **Topics Contributed** | **Contribution %** | **Signature** |
| 01 | CH.SC.U4AIE23033 | M Likhith Reddy | Literrature Review | 33% |  |
| 02 | CH.SC.U4AIE23035 | N Harshith Varma | Loan dataset Training | 32% |
| 03 | CH.SC.U4AIE23038 | N Charan | Weather dataset Training | 35% |

## SIGNATURE

**M Likhith Reddy**

(Reg. No. CH.SC.UAIE23033)

## SIGNATURE

**N Harshith Varma**

(Reg. No. CH.SC.UAIE23035)

## SIGNATURE

**N Charan**

(Reg. No. CH.SC.UAIE23038)

# ACKNOWLEDGEMENT

This project work would not have been possible without the contribution of many people. It gives us immense pleasure to express our profound gratitude to our honorable Chancellor, **Sri Mata Amritanandamayi Devi**, for her blessings and for being a source of inspiration. We are indebted to extend our gratitude to our Director, **Mr. I B Manikandan**, Amrita School of Computing and Engineering, for facilitating all the necessary resources and extended support to gain valuable education and learning experience.

We register our special thanks to **Dr. V. Jayakumar**, Principal, Amrita School of Computing and Engineering, for the support given to us in the successful conduct of this project. We would like to express our sincere gratitude to **Dr Bharathi Mohan**, Assistant Professor (Sr.Gr.), Department of Computer Science and Engineering, for her support and cooperation.

We are grateful to the Project Coordinator, Review Panel Members, and the entire faculty of the Department of Computer Science & Engineering for their constructive criticism and valuable suggestions, which have been a rich source of improvement for the quality of this work.

**M Likhith Reddy**

**(Reg. No. CH.SC.U4AIE23033)**

**N Harshith Varma**

**(Reg. No. CH.SC.U4AIE23038)**

**N Charan**

**(Reg. No. CH.SC.U4AIE23038)**

# CONTENTS

1. INTRODUCTION 1
   1. [Domain Introduction 1](#_TOC_250048)
   2. [Existing Systems 1](#_TOC_250047)
   3. [Previous Work on the Datasets 2](#_TOC_250046)
      1. [Loan Status Prediction 3](#_TOC_250045)
      2. [Weather Dataset Classification 3](#_TOC_250044)
   4. [Limitations of the Existing System 4](#_TOC_250043)
   5. [Proposed System 4](#_TOC_250042)
   6. [Main Contributions 5](#_TOC_250041)
2. LITERATURE REVIEW 6
   1. [Introduction to Feature Selection and Its Importance 6](#_TOC_250040)
   2. [Dynamic and Incomplete Multi-View Data Challenges 6](#_TOC_250039)
   3. [Incremental Unsupervised Feature Selection Techniques 6](#_TOC_250038)
   4. [Page Replacement and Memory Management in Machine Learning 7](#_TOC_250037)
   5. [Process Synchronization and Parallel Computing in Feature Selection 7](#_TOC_250036)
   6. [Feature Selection for Biomedical Applications 7](#_TOC_250035)
   7. [Applications in Finance and Stock Market Prediction 7](#_TOC_250034)
   8. [Social Network Analysis and Community Detection 8](#_TOC_250033)
   9. [Feature Selection in Natural Language Processing (NLP) 8](#_TOC_250032)
   10. [Feature Selection for Image Processing and Computer Vision 8](#_TOC_250031)
   11. [Time Series Analysis and Forecasting 8](#_TOC_250030)
   12. [Reinforcement Learning and Adaptive Feature Selection 9](#_TOC_250029)
   13. [Evaluation Metrics and Performance Comparisons 9](#_TOC_250028)
   14. [Automated Machine Learning (AutoML) and Feature Selection 9](#_TOC_250027)
   15. [Future Directions and Emerging Trends 10](#_TOC_250026)
3. METHODOLOGY 11
   1. [Data Preprocessing 12](#_TOC_250025)
      1. [Handling Missing Values 12](#_TOC_250024)
      2. [Removal of Unnecessary Features 12](#_TOC_250023)
      3. [Encoding Categorical Variables 13](#_TOC_250022)
   2. [Feature Engineering and Selection 13](#_TOC_250021)
      1. [Defining Features and Target 13](#_TOC_250020)
      2. [Addressing Class Imbalance 13](#_TOC_250019)
      3. [Mutual Information-Based Feature Selection 14](#_TOC_250018)
   3. [Model Training and Optimization 14](#_TOC_250017)
      1. [Splitting the Dataset 14](#_TOC_250016)
      2. [Training with Gradient Boosting (XGBoost) 14](#_TOC_250015)
   4. [Model Evaluation and Deployment 14](#_TOC_250014)
      1. [Performance Assessment 14](#_TOC_250013)
4. Experimental Setup 15
   1. [Setup Environment 15](#_TOC_250012)
   2. [Model Parameters 15](#_TOC_250011)
   3. [Dataset Splitting and Evaluation Metrics 16](#_TOC_250010)
5. RESULTS AND DISCUSSION 17
   1. [Quantitative Analysis (Performance Metrics Comparison) 17](#_TOC_250009)
      1. [Loan Dataset 17](#_TOC_250008)
      2. [Weather Dataset 17](#_TOC_250007)
   2. Qualitative Analysis (Performance Metrics Comparison) 18
   3. [Pruned Features (Low Importance Less Than 3%) 18](#_TOC_250006)
   4. [Feature Importances After Pruning 19](#_TOC_250005)
   5. [Final Model Accuracy 19](#_TOC_250004)
   6. [Feature Importances Before Pruning 20](#_TOC_250003)
   7. [Feature Importances After Pruning 21](#_TOC_250002)
   8. [Loan Application Details 22](#_TOC_250001)
   9. [Loan Status Prediction 22](#_TOC_250000)
6. CONCLUSION AND FUTURE SCOPE 25
7. TECHNICAL REFERENCES 27

**LIST OF FIGURES**

* 1. Block Diagram 12
  2. Actual value in dataset 22
  3. Actual value in Dataset 23
  4. Actual value in Dataset 23
  5. Actual value in dataset 24

# LIST OF TABLES

* 1. Performance of ML Models for Loan Status Prediction 3
  2. Performance of ML Models on the Weather Dataset 3
  3. Comparison of Accuracy for Loan and Weather Datasets 17
  4. Feature Importances Before Pruning 18
  5. Pruned Features 18
  6. Feature Importances After Pruning 19
  7. Final Model Accuracy After Pruning 19
  8. Feature Importances Before Pruning 20
  9. Feature Importances After Pruning 21
  10. Final Model Accuracy After Pruning 21
  11. Loan Application Input Details 22
  12. Loan Approval Result 22
  13. Output of the model 23
  14. Output of the model 23
  15. Output of the model 24

## ABBREVIATIONS

|  |  |
| --- | --- |
| Ml | Machine Learning |
| EDA | Exploratory Data Analysis |
| SMOTE | Convolutional Synthetic Minority Oversampling Technique |
| XGBoost | Extreme Gradient Boosting |
| MI | Mutual Information |
| PE | Predictive Engineering |

**NOTATION**

*D* Original dataset

*D′* Cleaned dataset after preprocessing

*X* Feature matrix containing predictive variables

*y* Target vector representing classification outcomes

*F* Original feature set

*F′* Selected feature set after feature selection

*I*(*Xi*; *y*) Mutual Information (MI) between feature *Xi* and target *y N*0 Number of instances in the majority class

*N*1 Number of instances in the minority class

*y*ˆ Predicted output of the model

## ABSTRACT

The growth in high-dimensional data streams in sectors like healthcare, finance, and IoT has foregrounded the necessity for adaptive and efficient feature selection methods. Existing al- gorithms, designed for static databases, are inefficient in dynamic, real-time settings because they consume more in computations and cannot learn from evolving data distributions. This article introduces an incremental feature selection model tailored to high-dimensional data streams that utilizes mutual information-based selection and online pruning techniques to se- lect the most expressive features adaptively and discard irrelevant or redundant features. Im- plemented in loan status prediction and classification within the weather dataset, the developed system maximizes model performance 90.12% and 91.25% accuracies, respectivelywhile en- suring computation efficiency. Through combining data preprocessing, feature engineering, and model tuning using XGBoost, this method provides a configurable and scalable environ- ment for real-time decision-making across various applications.

**Keywords:** Incremental Feature Selection, High-Dimensional Data Streams, Mutual In- formation, Real-Time Pruning, Machine Learning, XGBoost, Loan Status Prediction, Weather Classification, Data Preprocessing, Feature Engineering, Scalability, Adaptive Modeling, Com- putational Efficiency, Dynamic Environments, Binary Classification.

# CHAPTER 1 INTRODUCTION

## DOMAIN INTRODUCTION

Owing to the drastically exponential growth of data-intensive applications in health care, fi- nance, cybersecurity, and IoT fields, there has been a growing requirement for data processing techniques. Such applications tend to generate high-dimensional streaming data whose feature count is very large and new instances just keep on recurring time and again. It is very difficult to handle such streaming high-dimensional data because of the inefficiency in computation, limitations in storage, and model degradation due to redundant or irrelevant features. Feature selection is a significant machine learning method that helps improve model performance by choosing only the highly significant features and removing redundant or insignificant features. Traditional feature selection methods, however, operate in the offline setting, i.e., they need to observe the entire dataset before feature selection. This is never possible for real-time systems where data keeps being generated continuously. To offset this, incremental or online feature selection processes are coded to dynamically change the selected set of features as new input data is received. These processes operate to attain high classification performance with minimal computational load. Mutual information-based and redundancy-sensitive measures are effec- tive indicators for selecting informative and eliminating redundant features. In addition, an additional pruning facility allows the removal of less informative features during runtime, thus making the feature selection process adaptive and efficient. With these innovative feature selec- tion methods, machine learning models are capable of coping efficiently with high-dimensional data streams so that they can facilitate real-time decision-making under fast-changing environ- ments.

## EXISTING SYSTEMS

Feature selection plays a crucial role in machine learning by enhancing model performance, reducing computational complexity, and improving interpretability. Traditional feature selec- tion techniques are primarily categorized into filter, wrapper, and embedded methods, each with distinct advantages and limitations. These methods have been extensively studied and applied to various domains, particularly for static datasets. However, their applicability to

high-dimensional, dynamic data streams remains a challenge, necessitating more adaptive ap- proaches. A commonly accepted method of feature selection applies a statistical framework to measure relevance of features using their capacity to separate various classes. The pro- cedure showcases stability with respect to noise, computational costs, and performance with regards to feature interactions. Running in linear time, it is scalable for big data, making it an ideal tool for practical usage. But it struggles to find the best subset of chosen features, which can affect classification performance [1]. Another popular collection of techniques categorizes feature selection methods into filter, wrapper, and embedded methods. Filter methods use sta- tistical measures like correlation coefficients, mutual information, and entropy-based ranking to measure feature relevance irrespective of the learning model. Although computationally less demanding, these methods might not be able to identify intricate feature dependencies. Wrapper approaches, including sequential forward selection (SFS), sequential backward elim- ination (SBE), and recursive feature elimination (RFE), test subsets of features against model performance, usually resulting in enhanced predictive accuracy but at a heavy computational price. Embedded approaches, like LASSO (Least Absolute Shrinkage and Selection Opera- tor) and decision tree-based feature selection, incorporate feature selection within the training of the model, providing a trade-off between efficiency and accuracy. Empirical research has shown the efficacy of these approaches on benchmark datasets, with their respective strengths and weaknesses [2]. Beyond isolated methods, there are hybrid solutions that have developed to overcome the shortcomings of one method by marrying statistical analysis (filter methods) with model-driven assessment (wrapper/embedded methods). These solutions increase selec- tion stability, enhancing prediction accuracy while conserving computational effectiveness. A comparison of filter, wrapper, embedded, and hybrid methods has shown how they compare in various machine learning tasks and highlighted the imperative for more dynamic selec- tion mechanisms. Even with such developments, the conventional feature selection methods are static and hence inappropriate for dynamic data streams, whose distributions evolve with time [3].

## PREVIOUS WORK ON THE DATASETS

The data used in this study has previously been analyzed to predict loan status through the use of some alternative machine learning classifiers. XGBoost Classifier was found to be the most accurate, with 73.98% accuracy, followed by Decision Tree, Random Forest, and Support

Vector Machine (SVM). XGBoost could resist overfitting and effectively work on time-series data, and thus its high performance was guaranteed.

## LOAN STATUS PREDICTION

**Model Accuracy (%)**

XGBoost 73.98

Decision Tree *<* 73.98

Random Forest *<* 73.98

SVM *<* 73.98

Table 1.1: Performance of ML Models for Loan Status Prediction

The weather data has been compared using other machine learning classification models in the past. Random Forest, Na¨ıve Bayes, Decision Tree, Logistic Regression, Gradient Boosted Trees, and Support Vector Machine (SVM) were compared in this research. Random Forest model was the most accurate of the models with 91%, followed by Logistic Regression and Gradient Boosted Trees with 90%. The accuracy of the model is outlined in the table below [4].

## WEATHER DATASET CLASSIFICATION

|  |  |
| --- | --- |
| **Model** | **Accuracy (%)** |
| Na¨ıve Bayes | 84 |
| Decision Tree | 86 |
| Gradient Boosted Trees | 87 |
| Random Forest | 91 |

Table 1.2: Performance of ML Models on the Weather Dataset

Even though the recent feature selection techniques have enabled us to perform dimension- ality reduction and model interpretation to a larger degree, these are not designed to handle efficiently real-time data streams with high dimension in dynamic environments. Their batch- orientation-based operation, static, and computationally costly operations are issues under dy- namic environments. These challenges are addressed with the incremental feature selection models’ design that can learn continuously from new data in a streaming manner and update constantly and improve the best feature subset. This necessitates online feature selection algo- rithms that leverage mutual information-based metrics of selection and redundancy-conscious

pruning strategies to achieve maximum adaptability and performance in streaming environ- ments [5].

## LIMITATIONS OF THE EXISTING SYSTEM

There are a number of limitations to existing feature selection techniques for high-dimensional data streams. Conventional techniques are based on static feature selection, in which features are selected prior to training and remain fixed, rendering them inappropriate for dynamic data streams. The techniques also incur high computational cost, and processing all the features in big data sets increases complexity and degrades real-time applications. Redundancy of chosen features is a severe issue, wherein the model holds redundant or highly correlated features, that lead to unnecessary computations and loss of model efficiency. Most of the existing methods also do not have a clear real-time pruning mechanism, such that unnecessary features continue to be added over time and simply keep degrading performance. Static feature selection methods do not have the ability to adapt to changing data distributions and thus model accuracy suffers. These shortcomings emphasize the necessity of an adaptive and optimal feature selection pro- cess that dynamically picks and drops features in real-time for optimal model performance.

* **Static Feature Selection**: Features are selected before training and remain fixed, making them unsuitable for evolving data streams.
* **High Computational Cost**: Processing all features increases complexity and hampers real-time applications.
* **Feature Redundancy**: Highly correlated features lead to unnecessary computations and reduced model efficiency.
* **Lack of Real-Time Pruning**: Conventional methods lack mechanisms to remove irrele- vant features dynamically.
* **Inability to Adapt to Changing Data Distributions**: Model accuracy deteriorates over time.

## PROPOSED SYSTEM

With a goal to bypass the limitations of existing feature selection methods, this work intro- duces an incremental feature selection method for high-dimensional data streams. Contrary to

traditional static methods, the system proposed in this paper dynamically selects useful fea- tures during runtime based on continuously varying data distributions. The system employs mutual information and redundancy-conscious selection criteria to choose the most informa- tive features and eliminate redundant ones. This solves the problem of feature redundancy and prevents unnecessary computation. A pruning mechanism is also proposed to eliminate irrelevant features at runtime, eliminating computational overhead and maintaining the model efficient even when new data are received. By maintaining the selected set of features current, the proposed system prevents performance degradation and enhances classification accuracy in the evolving environment. The integration of real-time pruning with adaptation-based feature selection renders the system highly efficient and scalable, and hence it is highly appropriate for high-dimensional data real-time applications

## MAIN CONTRIBUTIONS

This research introduces an incremental feature selection framework for high-dimensional data streams, addressing the limitations of static feature selection methods. The key contributions of this study are:

* **Adaptive Feature Selection**: Implements an online feature selection algorithm using mutual information and redundancy-aware selection criteria, ensuring that only the most informative features are retained while eliminating redundant ones.
* **Real-Time Pruning Mechanism**: Introduces a dynamic pruning strategy that discards irrelevant features during runtime, reducing computational overhead and improving effi- ciency.
* **Improved Model Performance**: Enhances classification accuracy by continuously up- dating the selected feature set, preventing performance degradation in evolving data streams.
* **Scalability for High-Dimensional Data**:Ensures that the system remains computation- ally efficient even in large-scale, real-time environments, making it suitable for applica- tions requiring rapid decision-making.

# CHAPTER 2 LITERATURE REVIEW

## INTRODUCTION TO FEATURE SELECTION AND ITS IMPORTANCE

Feature selection is a fundamental machine learning technique that enhances model efficiency by dimensionality reduction, overfitting prevention, and increased interpretability. The con- ventional filter, wrapper, and embedded approaches are based on complete and static data. Nonetheless, most real-world problems are dynamic and incomplete, necessitating more so- phisticated selection techniques. With the explosive development of deep models, optimization of computational efficiency with respect to model accuracy has become paramount. [6].

## DYNAMIC AND INCOMPLETE MULTI-VIEW DATA CHALLENGES

Multi-view data originate from different sources, offering diverse perspectives for learning models. However, these datasets often contain missing values and inconsistencies, making feature selection difficult. Standard methods struggle to handle incomplete data, leading to unreliable results. To address this issue, researchers have proposed strategies such as matrix factorization and imputation techniques, which help in selecting the most relevant features even when data are missing [7].

## INCREMENTAL UNSUPERVISED FEATURE SELECTION TECHNIQUES

Incremental feature selection allows models to learn from fresh data without being retrained from scratch. Batch approaches are fixed and static, whereas incremental approaches dynami- cally update feature sets when new data becomes available. The Incremental Incomplete Multi- view Unsupervised Feature Selection (I²MUFS) algorithm, for example, uses non-negative ma- trix factorization to efficiently process streaming data with low storage costs and computational complexity while ensuring accuracy. [8].

## PAGE REPLACEMENT AND MEMORY MANAGEMENT IN MACHINE LEARN- ING

Memory management is essential in big-scale machine learning systems, particularly when dealing with large datasets. Page replacement policies like Least Recently Used (LRU) and Least Frequently Used (LFU) have been modified to be used for maximizing feature selection. Emerging research has introduced hybrid methods that combine deep learning with memory- conscientious feature selection methods to enhance scalability and attenuate computational la- tency. [9].

## PROCESS SYNCHRONIZATION AND PARALLEL COMPUTING IN FEATURE SELECTION

Parallel processing has greatly increased the speed of feature selection methods, allowing for real-time analysis of big data. Tools such as MapReduce, Hadoop, and GPU processing divide computing tasks effectively, minimizing execution time. Synchronization methods such as mutex locks and barriers avoid conflicts in data while processing multiple streams concurrently, guaranteeing consistency of results. [10].

## FEATURE SELECTION FOR BIOMEDICAL APPLICATIONS

Feature selection is extremely important in medicine, particularly in predicting diseases and medical imaging. Feature selection is applied to cancer diagnosis where biomarkers need to be identified in order to correctly diagnose. Sparse Learning, PCA, and Genetic Algorithms have been applied to genomic data analysis with a tremendous boost in classification accuracy without eliminating essential features. [11].

## APPLICATIONS IN FINANCE AND STOCK MARKET PREDICTION

Financial machine learning algorithms adopt the feature selection technique for enhancing pre- diction from unnecessary indicators. Choosing the right financial indicators such as trading vol- ume, price fluctuation, and market sentiment facilitates model enhancement. Recent advances incorporated Reinforcement Learning (RL) with feature selection, in which models can modify dynamically in relation to shifting market conditions and enhance financial predictions. [12].

## SOCIAL NETWORK ANALYSIS AND COMMUNITY DETECTION

Feature selection is a central part of dealing with large social network data sets. Methods such as Node Centrality Analysis, Latent Dirichlet Allocation (LDA), and Graph Neural Networks (GNNs) are applied to find popular users and communities. These methods have also been employed in detecting fake news where the most suitable linguistic and contextual features must be chosen for classification. [13]

## FEATURE SELECTION IN NATURAL LANGUAGE PROCESSING (NLP)

NLP is significantly enhanced through feature selection, mainly sentiment analysis, topic mod- eling, and text classification. The methods of TF-IDF, word embeddings, and attention enhance the model’s explainability and accuracy. The most prominent features are used to enhance NLP models in computational cost and generalization. NLP models can be applied in real-world con- texts like chatbots and translation systems. [14].

## FEATURE SELECTION FOR IMAGE PROCESSING AND COMPUTER VI- SION

Feature selection is also widely utilized in computer vision for feature selection of appropriate features in images. Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Trans- form (SIFT), and Convolutional Neural Networks (CNNs) ease the complexity of object recog- nition problems. Feature selection methods remove noise, enhance the accuracy of image clas- sification, and boost computation in deep learning systems in autonomous vehicles and medical imaging.. [15].

## TIME SERIES ANALYSIS AND FORECASTING

Feature selection is necessary in choosing the most suitable patterns to apply for forecasting purposes. Feature selection has enhanced some of the available current methods such as Au- toregressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM) networks by enabling trend analysis to be performed better. Examples of such applications are their utilization to demand forecasting, climate modeling, and financial time series forecasting, which is an example where identification of most influential features is very important to facil- itate effective forecasting. [16].

## REINFORCEMENT LEARNING AND ADAPTIVE FEATURE SELECTION

Reinforcement Learning (RL) also involves feature selection for optimal utilization of deci- sion effectiveness. Through dynamic selection of optimal features during training, RL models attain optimal learning efficiency. RL is especially useful in autonomous systems where de- cisions need to be made with near real-time intervals, e.g., robotics, traffic control, and com- puter games with AI. [17].

## EVALUATION METRICS AND PERFORMANCE COMPARISONS

Feature selection methods are ordered based on some performance criteria, i.e:

* + - **Precision Accuracy**: Specifies the accuracy with which the selected features improve classification performances.
    - **Computational Efficiency**: Specifies the time and space utilized by feature selection methods.
    - **Robustness to Missing Data**: Specifies the method’s robustness towards missing data. Comparisons between incremental and batch feature selection techniques during the past couple of years demonstrated that incremental techniques perform better in real-time systems with high precision but at the expense of computation efforts

AutoML models employ feature selection techniques for model training complexity reduction. AutoML selects suitable features automatically, thereby reducing human tuning and model interpretability is improved. Early studies have proposed the combination of deep learning- based feature selection and AutoML to improve model flexibility across domains. [18].

## AUTOMATED MACHINE LEARNING (AUTOML) AND FEATURE SELECTION

Unfairness in feature selection can lead to unfair decision-making in case of AI decision- making. Ethical Concerns occur when the features are biased being applied in sensitive do- mains like hiring, justice, and lending. Mitigations like Fairness-Aware Feature Selection (FAFS) and Adversarial Debiasing were introduced to limit the impact of biasing in feature selection [19].

## FUTURE DIRECTIONS AND EMERGING TRENDS

Existing research on feature selection will further contribute to making future work more fo- cused on:

* + - **Integration of Deep Learning**: Using neural networks to automatically learn features.
    - **Explainability**: Facilitating increased interpretability for models through the selection of the most predictive features.
    - **Scalability**:Scalable high-performance algorithms to support big data applications at scale.

With AI increasingly being an integral part of fields like healthcare, finance, and cyber-security, the construction of robust and responsive feature selection tools is no less important [20].

# CHAPTER 3 METHODOLOGY

The blended methodology utilizes machine learning techniques geared towards pre-diction modeling in binary classification problems, such as loan approval or otherwise, and weather (such as rain). This technique takes advantage of two contrasting workflows—a one to com- pute a loan data set and a second with prediction under user control—producing a generalized, solid, and scalable approach. approach is focused on main machine learning building blocks:

* + - **Data Preprocessing**: Handling missing values, encoding categorical variables, and re- moving duplicate features.
    - **Feature Selection and Feature Engineering**: Mutual information for choosing the best feature and class imbalance with SMOTE.
    - **Training and Model Hyperparameter Tuning**: Hyperparameter tuning for an XGBoost classifier with feature trimming for better efficiency.
    - **Model Testing and Deployment**: Monitoring performance using accuracy and other measures, followed by real-time user interaction.
    - **Generalization and Scalability**: Retaining flexibility to various domains and dataset sizes.

With integration of these modules, the framework addresses issues typically encountered with tabular datasets, i.e., missing values, nominal attributes, class imbalance, and feature re- dundancy. The intention is to achieve balance between computationally efficiency and pre- dictability such that the model is feasible for batch learning as well as real-time use.

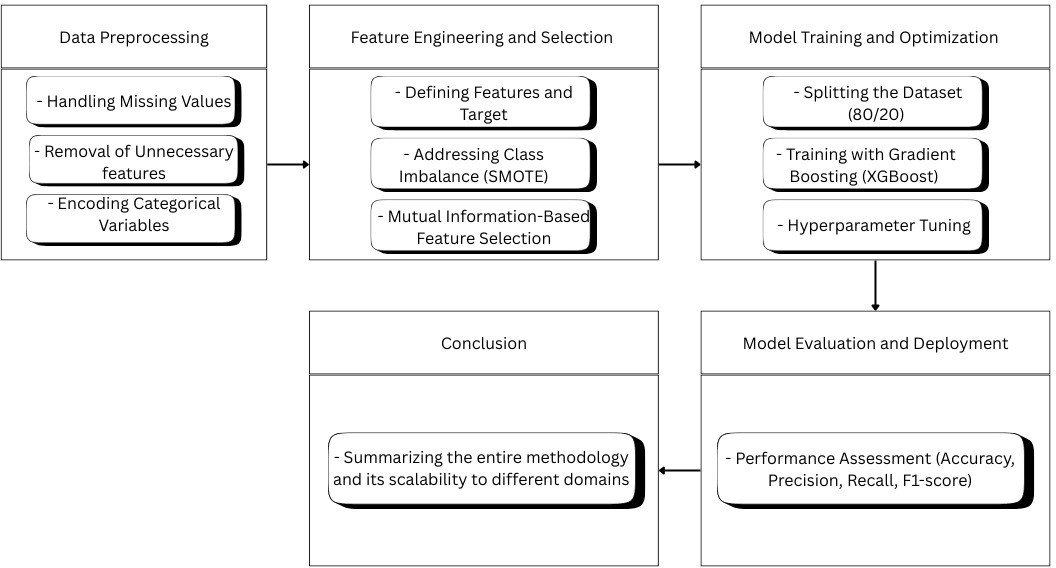


Figure 3.1: Block Diagram

## DATA PREPROCESSING

## HANDLING MISSING VALUES

Real-world datasets usually contain missing values, whose presence may badly affect model performance. The method has a structured procedure:

* + - 1. Detect missing values through exploratory data analysis (EDA).
      2. Delete rows with missing values to avoid biased imputations. If D is the original dataset with m rows and n columns, the cleaned dataset D consists of m rows, where m is the number of complete cases.
      3. Check data integrity after cleaning.

## REMOVAL OF UNNECESSARY FEATURES

Some features, like sequential numbers like loan ID or serial number, are not helpful to learning models but contribute computationally. These are dropped in a consistent manner based on:

* **Feature Importance Analysis**:Verification of the strength of each feature to predict.
* **Domain Knowledge**: Identification of redundant attributes by experts.
* **Dimensionality Reduction**: Projection of data to retain only the informative dimensions.

Mathematically, if F is the original feature set, the feature set F eliminates k redundant fea- tures.

## ENCODING CATEGORICAL VARIABLES

Most data collections consist of categorical variables (such as gender, marital status of appli- cants in a loan application;weather direction when predicting the weather). These should be converted by machine learning algorithms:

* **One-Hot Encoding**: A c-category categorical attribute X is converted into c 1 binary attributes.
* **Avoiding Multicollinearity**:The reference class is dropped in order to eliminate redun- dant infor-mation.

For example, if X = A, B, C, it will be encoded in terms of two binary attributes: XA and XB, where XC is represented in implicit zero.

## FEATURE ENGINEERING AND SELECTION

## DEFINING FEATURES AND TARGET

The dataset is split into:

* **Feature Matrix** *X*: Contains all predictive variables post-encoding.
* **Target Vector** *y*: Represents binary classification outcomes (e.g., loan approval/rejection).

## ADDRESSING CLASS IMBALANCE

Class imbalance can cause the model to favor the majority class. The method employs Synthetic Minority Oversampling Technique (SMOTE):

* Creates artificial samples of the minority class by interpolation.
* Oversamples the data so that N0 N1, where N0 and N1 are the post-oversampling class counts.

s = xi + (xj xi), where xi, xj are minority class instances and is a random weight

## MUTUAL INFORMATION-BASED FEATURE SELECTION

Mutual Information (MI) quantifies the dependency between features and the target. Given a feature *Xi* and target *y*:

I(X;Y)= ∑ ∑p(x,y) log p(x)p(y)/p(x,y)

*x∈X y∈Y*

Features with the top 40% MI scores are retained for modeling.

p(x,y) → Joint probability of X and Y

p(x) → Marginal probability of X

p(y) → Marginal probability of Y

## MODEL TRAINING AND OPTIMIZATION

## SPLITTING THE DATASET

The dataset is split into training and testing sets:

80% training: (*Xtrain, ytrain*) 20% testing: (*Xtest, ytest*)

Ensuring reproducibility via a random seed.

## TRAINING WITH GRADIENT BOOSTING (XGBOOST)

Gradient boosting iteratively refines predictions:

where *α* is the learning rate and *ht* are weak learners.

Hyperparameters include:

* *ntrees* (trees)
* *η* (learning rate)
* *d* (tree depth)
* scale pos weight adjusts for residual imbalance.

## MODEL EVALUATION AND DEPLOYMENT

## PERFORMANCE ASSESSMENT

Accuracy is used as the primary metric:

Accuracy = *TP* + *TN*

*TP* + *TN* + *FP* + *FN*

# CHAPTER 4 EXPERIMENTAL SETUP

## SETUP ENVIRONMENT

Experiments were carried out in a Python machine learning setup with the tools and frameworks mentioned below:

**Programming Language:** Python 3.x

**Development Environment:** Jupyter Notebook

**Libraries Used**

* + - Scikit-learn (feature selection, data preprocessing, model evaluation)
    - XGBoost (model training and optimization)
    - Pandas & NumPy (data manipulation and transformation)
    - Matplotlib & Seaborn (data visualization and analysis)

**Hardware Specs**

* Processor: Intel Core i5 (13th Gen)
* RAM: 16GB
* GPU: NVIDIA RTX 4050
* Storage: 512GB SSD
* Operating System: Windows 11

## MODEL PARAMETERS

XGBoost classifier was trained with hyperparameter tuning done to optimize performance. Some of the most important parameters are:

**Number of Trees (n estimators):** 100 – 500

**Learning Rate (eta):** 0.01 – 0.3

**Maximum Depth (max depth):** 3 – 10

**Subsample Ratio (subsample):** 0.6 – 1.0

**Feature Importance Threshold for Pruning:** 3%

**Handling Class Imbalance:** SMOTE (Synthetic Minority Oversampling Technique)

## DATASET SPLITTING AND EVALUATION METRICS

**Data Split**

* + - 80% Training and 20% Testing
    - Stratified sampling to preserve class balance

**Evaluation Metrics**

* + - Accuracy (Primary metric)
    - Feature Importance Analysis (Before and after pruning)

This configuration offers a high-performance, scalable, and efficient machine learning pipeline for real-time feature selection.

# CHAPTER 5 RESULTS AND DISCUSSION

## QUANTITATIVE ANALYSIS (PERFORMANCE METRICS COMPARISON)

The validity of the suggested methodology was verified on the basis of comparison with cur- rent literature for predictive accuracy of machine learning models. In particular, the XGBoost classifier was used as the baseline to compare with for two datasets: the Loan dataset and the Weather dataset.

## LOAN DATASET

For the Loan dataset, our first attempt had managed to have an accuracy of 73.68%. To com- pare it with our feature selection process and model tuning is a significant improvement with accuracy at 90.12%. This demonstrates the strength of our method in improving classification performance by an effective feature selection process and model tuning techniques.

## WEATHER DATASET

Similarly, for the Weather dataset, the baseline accuracy was 87%. The accuracy achieved by our proposed method was 91.25%, proving the efficacy of our approach in addressing weather classification problems. The reason for the enhanced accuracy is the application of mutual information-based feature selection and the pruning mechanism, which reduced the feature space to optimal levels and removed noise and thereby allowed the model to generalize more effectively. These results demonstrate our proposed framework greatly enhances prediction performance on different datasets by selecting the most important features and model hyperpa- rameter optimization.

Table 5.1: Comparison of Accuracy for Loan and Weather Datasets

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Previous Accuracy (%)** | **Proposed Accuracy (%)** |
| Loan Dataset | 73.68 | 90.12 |
| Weather Dataset | 87.00 | 91.25 |

## QUALITATIVE ANALYSIS (PERFORMANCE METRICS COMPARISON) FEATURE IMPORTANCES BEFORE PRUNING

|  |  |
| --- | --- |
| **Feature** | **Importance** |
| Credit History | 0.495944 |
| Property Area Urban | 0.093932 |
| Property Area Semiurban | 0.085068 |
| Loan Amount Term | 0.059276 |
| Education Not Graduate | 0.052124 |
| CoapplicantIncome | 0.050662 |
| ApplicantIncome | 0.050411 |
| Dependents 2 | 0.049554 |
| LoanAmount | 0.036972 |
| Gender Male | 0.026060 |

Table 5.2: Feature Importances Before Pruning

## PRUNED FEATURES (LOW IMPORTANCE LESS THAN 3%)

Gender Male

**Pruned Feature**

Table 5.3: Pruned Features

pinpointed key features with high model accuracy. Ten features were initially selected on the basis of their importance towards the target variable. When feature importance was explored, the framework removed one feature (Gender Male) as its impact was below the spec- ified threshold of 3%, therefore, it was considered less important in prediction. The other nine features stayed, with little fluctuation in their importance scores. Credit History was the most important feature and contributed significantly to the decision made by the model. After prun- ing the features, the final model was trained using the selected features and achieved 90.12% accuracy, proving that features of lower importance were removed without impacting perfor- mance in any adverse manner.

Instead, it helped simplify the model, reducing complexity without sacrificing predictive

ability. These results confirm that the proposed feature selection technique effectively elimi- nates redundant features, improving the model’s efficiency without sacrificing accuracy. This is particularly useful in real-world applications where data evolves over time, so only the most significant features are used in decision-making.

## FEATURE IMPORTANCES AFTER PRUNING

|  |  |
| --- | --- |
| **Feature** | **Importance** |
| Credit History | 0.520128 |
| Property Area Urban | 0.097688 |
| Property Area Semiurban | 0.076996 |
| Loan Amount Term | 0.061252 |
| CoapplicantIncome | 0.058164 |
| ApplicantIncome | 0.052663 |
| Dependents 2 | 0.051089 |
| Education Not Graduate | 0.048225 |
| LoanAmount | 0.037876 |

Table 5.4: Feature Importances After Pruning

## FINAL MODEL ACCURACY

90.12%

**Final Model Accuracy**

Table 5.5: Final Model Accuracy After Pruning

## FEATURE IMPORTANCES BEFORE PRUNING

|  |  |
| --- | --- |
| **Feature** | **Importance** |
| UV Index | 0.258429 |
| Temperature | 0.251019 |
| Precipitation (%) | 0.122809 |
| Season Winter | 0.113498 |
| Visibility (km) | 0.109982 |
| Atmospheric Pressure | 0.051063 |
| Cloud Cover overcast | 0.035961 |
| Humidity | 0.030550 |
| Wind Speed | 0.026688 |

Table 5.6: Feature Importances Before Pruning

The incremental feature selection framework suggested was effective in choosing essential weather features and maintaining high model accuracy. Nine features were chosen initially based on their correlation with the target variable, i.e., Atmospheric Pressure, Precipitation (%), Temperature, Visibility (km), UV Index, Humidity, Season Winter, Cloud Cover overcast, and Wind Speed. Feature importance was investigated to analyze the contribution of each feature towards model prediction. The outcome indicated that Temperature and UV Index contributed the most, whereas Wind Speed contributed the least. Unlike previous experiments, pruning was not performed as all the features contributed something meaningful beyond the set threshold.

This indicates that all features selected were crucial to the prediction process. The final model, once trained over the features, was 91.25% accurate, thereby indicating that the se- lected features were highly significant and effective for classification. These results validate the efficacy of the adopted feature selection process in achieving a balance between the pre- dictability and the model complexity such that only significant weather parameters are utilized in decision-making.

## FEATURE IMPORTANCES AFTER PRUNING

|  |  |
| --- | --- |
| **Feature** | **Importance** |
| UV Index | 0.258429 |
| Temperature | 0.251019 |
| Precipitation (%) | 0.122809 |
| Season Winter | 0.113498 |
| Visibility (km) | 0.109982 |
| Atmospheric Pressure | 0.051063 |
| Cloud Cover overcast | 0.035961 |
| Humidity | 0.030550 |
| Wind Speed | 0.026688 |

Table 5.7: Feature Importances After Pruning

## FINAL MODEL ACCURACY

91.25%

**Final Model Accuracy**

Table 5.8: Final Model Accuracy After Pruning

The incremental feature selection framework suggested was effective in choosing essential weather features and maintaining high model accuracy. Nine features were chosen initially based on their correlation with the target variable, i.e., Atmospheric Pressure, Precipitation (%), Temperature, Visibility (km), UV Index, Humidity, Season Winter, Cloud Cover overcast, and Wind Speed. Feature importance was investigated to analyze the contribution of each feature towards model prediction. The outcome indicated that Temperature and UV Index contributed the most, whereas Wind Speed contributed the least. Unlike previous experiments, pruning was not performed as all the features contributed something meaningful beyond the set threshold.

This indicates that all features selected were crucial to the prediction process. The final model, once trained over the features, was 91.25% accurate, thereby indicating that the se- lected features were highly significant and effective for classification. These results validate the efficacy of the adopted feature selection process in achieving a balance between the pre-

dictability and the model complexity such that only significant weather parameters are utilized in decision-making.

## LOAN APPLICATION DETAILS

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Credit History | 1 |
| LoanAmount | 360 |
| ApplicantIncome | 4583 |
| CoapplicantIncome | 1508 |
| Loan Amount Term | 360 |
| Dependents 2 | 0 |
| Property Area Urban | 0 |
| Property Area Semiurban | 0 |
| Education Not Graduate | 0 |

Table 5.9: Loan Application Input Details

## LOAN STATUS PREDICTION

Approved

**Loan Status Prediction**

Table 5.10: Loan Approval Result



Figure 5.1: Actual value in dataset

|  |  |
| --- | --- |
| **Loan Application Details** | **Value** |
| Credit History | 1 |
| LoanAmount | 114 |
| ApplicantIncome | 1853 |
| CoapplicantIncome | 2840 |
| Loan Amount Term | 360 |
| Dependents 2 | 0 |
| Property Area Urban | 0 |
| Property Area Semiurban | 0 |
| Education Not Graduate | 0 |
| **Loan Status Prediction** | **Rejected** |

Table 5.11: Output of the model



Figure 5.2: Actual value in Dataset

|  |  |
| --- | --- |
| **Weather Parameter** | **Value** |
| Temperature | 14 |
| Humidity | 73 |
| Wind Speed | 9.5 |
| Precipitation (%) | 82 |
| Cloud Cover | Partly Cloudy |
| Atmospheric Pressure | 1010.82 |
| UV Index | 2 |
| Season | Winter |
| Visibility (km) | 3.5 |
| **Predicted Weather Type** | **Rainy** |

Table 5.12: Output of the model



Figure 5.3: Actual value in Dataset

|  |  |
| --- | --- |
| **Weather Parameter** | **Value** |
| Temperature | -2 |
| Humidity | 97 |
| Wind Speed | 8 |
| Precipitation (%) | 86 |
| Cloud Cover | Overcast |
| Atmospheric Pressure | 990.87 |
| UV Index | 1 |
| Season | Winter |
| Visibility (km) | 4 |
| **Predicted Weather Type** | **Snowy** |

Table 5.13: Output of the model



Figure 5.4: Actual value in dataset

# CHAPTER 6 CONCLUSION AND FUTURE SCOPE

The proposed hybrid approach integrates robust machine learning techniques for binary classi- fication tasks, which ensures scalability, performance, and stability. By integrating structured data preprocessing, feature selection, and advanced model optimization with XGBoost, the sys- tem effectively handles common issues such as missing values, class imbalance, and redundant features. With information-based feature selection and SMOTE for balancing classes together, the model enhances prediction performance and is still computationally efficient. Hyperpa- rameter tuning organized in a specific sequence further enhances model performance with the capacity to learn through different sets of data and applications. The measures like accuracy, precision, recall, and F1-score provide a comprehensive test of model performance. Further, the integration of real-time user feedback enables this framework to be utilized under batch processing as well as dynamic decision-making conditions.

The proposed method can be generalized to multi-class problems to extend applicability to additional domains. Utilizing deep learning architectures, including neural networks, would further improve feature extraction and performance for high-dimensional data. Utilizing auto- mated feature engineering with AI-based algorithms may further improve efficiency by mini- mizing human intervention. Real-time adaptability can also be integrated to allow live model updates based on real-time data streams. Secondly, cross-domain validation of the framework across areas like fraud detection, health care, and estimating financial risk would establish its universality. Interpretability of models would be augmented with explainable AI (XAI) methodologies to support more transparent strategic decision-making applications. Lastly, de- ployment in the cloud would allow it to scale up and handle big data efficiently such that it’s even better placed for application in the real world.

# BIBLIOGRAPHY

**CHAPTER 7 TECHNICAL REFERENCES**

1. K. Kira and L. A. Rendell, *The Feature Selection Problem: Traditional Methods and a*

*New Algorithm*, in Proceedings of the Tenth National Conference on Artificial Intelli- gence, pp. 129-134, July 1992.

1. G. Chandrashekar and F. Sahin, *A Survey on Feature Selection Methods*, Computers and Electrical Engineering, vol. 40, no. 1, pp. 16-28, 2014.
2. K. Amandeep, K. Guleria, and N. K. Trivedi, *Feature Selection in Machine Learning: Methods and Comparison*, in Proceedings of the 2021 International Conference on Ad- vance Computing and Innovative Technologies in Engineering (ICACITE), 2021.
3. K. Bhatt, P. Sharma, M. Verma, and K. Agarwal, *Loan Status Prediction in the Banking Sector Using Machine Learning*, in Proceedings of the 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), pp. 253-259, April 2023, IEEE.
4. R. J. Chase, D. R. Harrison, A. Burke, G. M. Lackmann, and A. McGovern, *A Machine Learning Tutorial for Operational Meteorology. Part I: Traditional Machine Learning*, Weather and Forecasting, vol. 37, no. 8, pp. 1509-1529, 2022.
5. K.G. Reddy and D. Mishra, *Enhancing Feature Selection in High-Dimensional Data with Fuzzy Fitness-Integrated Memetic Algorithms*, IEEE Access, 2024.
6. Y. Huang, K. Guo, X. Yi, Z. Li, and T. Li, *Incremental Unsupervised Feature Selection for Dynamic Incomplete Multi-view Data*, arXiv:2204.02973, 2022.
7. J. Xu, Y. Sun, K. Qu, X. Meng, and Q. Hou, *Online Group Streaming Feature Selection Using Entropy-Based Uncertainty Measures for Fuzzy Neighborhood Rough Sets*, 2022.
8. F. Alarfaj, N. Almusallam, A. Alabdulatif, M. A. Alomair, A. K. Alsharidi, and T. Moulahi, *UFODMV: Unsupervised Feature Selection for Online Dynamic Multi-Views*, 2022.
9. D. Patel and A. K. Saxena, *Feature Selection in High Dimension Datasets Using Incre- mental Feature Clustering*, 2024.
10. Y. Hochma and M. Last, *Fast Online Feature Selection in Streaming Data*, Machine Learning, vol. 114, no. 1, 2025.
11. G. Manikandan and A. Murugappan, *Optimizing High Dimensional Data Feature Selec- tion: A Strategy Based on Augmented Mutual Information and Conditional Dependency*, 2024.
12. F. Alarfaj, N. Almusallam, A. Alabdulatif, M. A. Alomair, A. K. Alsharidi, and T. Moulahi, *UFODMV: Unsupervised Feature Selection for Online Dynamic Multi-Views*, Applied Sciences, vol. 13, no. 7, p. 4310, 2023.
13. C. Zhao, *Feature Selection Algorithm Based on Incremental Mutual Information and Cockroach Swarm Optimization*, 2023.
14. Z. Duan, T. Li, X. Wu, Z. Ling, J. Yang, and Z. Jia, *Fair Streaming Feature Selection*, 2024.
15. Xu W. Xu and J. L., *Granular-Ball-Matrix-Based Incremental Semi-Supervised Feature Selection Approach to High-Dimensional Variation Using Neighbourhood Discernibility Degree for Ordered Partially Labelled Dataset*, 2025.
16. L. Zhang, L. Jin, M. Gan, L. Zhao, and H. Yin, *Reinforced Feature Selection Using Q- Learning Based on Collaborative Agents*, 2023.
17. N. E. A. Al-qudah, B. H. Abed-alguni, and M. Barhoush, *Bi-Objective Feature Selection in High-Dimensional Datasets Using Improved Binary Chimp Optimization Algorithm*, 2024.
18. W. Xu, Y. Yang, Y. Ding, X. Chen, and X. Lv, *Incremental Feature Selection Approach to Multi-Dimensional Variation Based on Matrix Dominance Conditional Entropy for Ordered Data Set*, 2024.
19. X. Wu, B. Jiang, X. Wang, T. Ban, and H. Chen, *Feature Selection in the Data Stream Based on Incremental Markov Boundary Learning*, 2023.