

**AN ENSEMBLE-BASED ALGORITHM FOR EFFICIENT
CLASSIFICATION OF REAL TIME DATA STREAMS**

A thesis submitted to the
University of Petroleum and Energy Studies

For the Award of
Doctor of Philosophy
in
Computer Science and Engineering

By
MONIKA ARYA

December 2021

Supervisor(s)

Dr. Hanumat G. Sastry

Dr. Ashutosh Pashricha



School of Computer Science
University of Petroleum and Energy Studies
Energy Acres, P.O. Bidholi via Prem Nagar,
Dehradun, 248007: Uttarakhand, India.

**AN ENSEMBLE-BASED ALGORITHM FOR EFFICIENT
CLASSIFICATION OF REAL TIME DATA STREAMS**

A thesis submitted to the
University of Petroleum and Energy Studies

For the Award of
Doctor of Philosophy
in
Computer Science and Engineering

By
MONIKA ARYA

December 2021

Supervisor

Dr. Hanumat G. Sastry

Professor,

School of Computer Science,

University of Petroleum & Energy Studies

External- Supervisor

Dr. Ashutosh Pasricha

OFS Director,

Schlumberger, India



School of Computer Science

University of Petroleum and Energy Studies

Energy Acres, P.O. Bidholi, via Prem Nagar,

Dehradun, 248007: Uttarakhand, India

Declaration

I declare that the thesis entitled “**An ensemble-based algorithm for efficient classification of real time data streams**” has been prepared by me under the guidance of Dr. Hanumat G Sastry, Professor at School of Computer, University of Petroleum & Energy Studies and Dr. Ashutosh Pasricha, OFS director at Schlumberger India. No part of this thesis has formed the basis for the award of any degree or fellowship previously.



MONIKA ARYA

School of Computer Science,

University of Petroleum & Energy Studies,


Bidholi via Prem Nagar, Dehradun, UK, INDIA

DATE: 20th January 2022

Certificate

I certify that **Monika Arya** has prepared her thesis entitled “**An ensemble-based algorithm for efficient classification of real time data streams**”, for the award of the Ph.D. degree of the University of Petroleum & Energy Studies, under my guidance. She has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

Supervisor


Dr. Hanumat G Sastry
Professor
School of Computer Science
University of Petroleum & Energy Studies,
Bidholi, via Prem Nagar, Dehradun,
UK, INDIA
DATE:

CERTIFICATE

I certify that the thesis entitled “*An ensemble-based algorithm for efficient classification of real time data streams*” by **Monika Arya (SAP ID: 500042599)**, a research scholar at University of Petroleum & Energy Studies, Dehradun, submitted thesis in partial completion of the requirements for the award of the Degree of Doctor of Philosophy in School of Computer Science is an original work carried out by her under my supervision and guidance. To the best of my knowledge, it is certified that the work has not been submitted anywhere else for the award of any other diploma or degree of this or any other university.



External Supervisor

Dr. Ashutosh Pasricha

OFS Account Director

Schlumberger India

Abstract

The evolution of connected devices worldwide has resulted in the generation of massive amounts of data called the data streams. Data stream mining is the transformation of the data stream into valuable knowledge using data mining. It is an emerging topic among researchers due to its significance in many real-life applications in different areas like economy, business, health care, and scientific research etc. Data stream classification is a well-studied problem among other data stream analysis methods. Over the past decade, research community has developed various robust data stream classification techniques. However, some inherent difficulties continue to pose a challenge to both current-generation hardware and cutting-edge algorithmic solutions. Unbound size, changing speed, and uncertain data attributes of arriving instances from a data stream are a few examples of these issues. Data stream classifiers need to be adaptive and efficient to overcome the challenges associated with classifying dynamic data streams. Studies reveal that the deep learning (DL) approaches addresses these challenges efficiently and provide significant improvements over traditional ML approaches. DL model consist of multiple-layers and eliminate the need for using handcrafted and engineered feature sets in training. As a result, the models easily extract features that may not be obvious to the human eye. Furthermore, DL models boost accuracy. The main focus of the present research work is to study classification solutions for data streams and to build an efficient classification algorithm for data stream classification using ensemble and deep learning technique that fulfills the requirements of both binary and multiclass data streams.

The thesis presents a detailed introduction to data streams and the importance of data stream classification in the real world. Further, a detailed literature review finds some techniques proved better with binary data streams while others proved better

in multiclass data streams. It is also noticed that the existing approaches use metrics insufficient to evaluate the performance of data stream classifiers. Furthermore, it is observed that no technique exists in the literature to utilize the ensemble-based approach to optimize the deep learning model to classify data streams. And hence, the research proposes a novel framework for efficient data stream classification. The framework consists of three phases: learning phase for classifying data streams using DL model, optimization phase for optimizing the DL model using an extra tree ensemble-based approach and prediction phase for predicting the class of input data stream. The deep learning model consists of multiple layers and automatically discovers the essential relationships during the learning phase. The deep learning model is further optimized for better and accurate predictions using extra tree ensemble approach. In the extra tree ensemble-based optimization process, the newly arriving features in the data streams are incorporated in a feature subset for classification. Thus, the optimization process makes the framework adaptive for newly arriving features and hence improves the overall performance of the model. Further, the proposed Deep Ensemble Algorithm Learning (DEAL) framework is compared with recent and state-of-the-art works, and the results are encouraging. In a gist, the present research program, “An efficient ensemble-based classification algorithm for real-time data streams,” provides a novel framework for data stream classification. The proposed work is adaptable for latent patterns, handles model overfitting, and has good prediction as well as categorical accuracy. The model is evaluated using vital performance metrics. The results demonstrate its superiority over state-of-the-art algorithms as well as recent ones and improvement of categorical accuracy by approximately 22.5 percent.

Acknowledgment

The journey toward a PhD is a confluence of multiple learnings; my success in this journey has been made possible only because of the help received from many individuals and their continual advice and counsel. I am extremely grateful to all of them.

First, I express my deepest gratitude to my research supervisor, Dr. Hanumat G Sastry, for his unconditional support and motivation through the whole process. His constant guidance helped me not only in my research but also in real life. He gave me the opportunity to explore my research potential and taught me the ropes. His simplicity, positivity, and discipline have been an inspiration for me. I, therefore, have a sincere desire to emulate him in his humility and discipline in my life.

I am indebted to my external supervisor, Dr. Ashutosh Pasricha, for his warm reception and insightful discussions and instructions during my research study. His encouragement and guidance provided a good basis for this thesis. I express my heartfelt thanks to him for always supporting me patiently and for his encouragement.

I am extremely grateful to the distinguished authors whose precious works I consulted and referred to in my research work.

I wish to convey my appreciation to my friends and colleagues at the Bhilai Institute of Technology, Durg, especially Chaitali Choudhary, Vivek Parganiha, Sumit Sar, and Dr. Sunita Soni, for their constant support and motivation. I extend my sincere and heartfelt thanks to Anand Motwani, VNIT, Bhopal, for his much-needed support during my PhD journey. I also express my thanks to all whose names have not been

mentioned individually but who have, nevertheless, helped me directly or indirectly in this work.

I am blessed to have a very loving and supportive family. Words cannot express my appreciation and gratitude toward my husband, Dr. Ajay Arya, for his constant love, support, motivation, and patience during my research journey. I also thank my mother and elder brother, Akhilesh, who, through their blessings and constant efforts, have molded me into the person I am today. I am thankful to my two lovely kids, Akshara and Akshat, who have shown so much patience with and confidence in me.

Above all, I owe everything to Almighty God and thank him for granting me the wisdom, health, and strength to undertake this research task and for showering me with his blessings to complete my thesis successfully.

Monika Arya

Table of Contents

DECLARATION.....	i
CERTIFICATE.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	v
TABLE OF CONTENT.....	vii
LIST OF ABBREVIATIONS.....	x
LIST OF FIGURES.....	xii
LIST OF TABLES.....	xiv
LIST OF PUBLICATIONS.....	xv
Chapter 1 Introduction and Motivation.....	1
1.1 Data streams and Data stream mining.....	3
1.2 Different approaches for data stream classification.....	8
1.2.1 Tree-based approach.....	9
1.2.2 Rule-based approach.....	9
1.2.3 Ensemble-based approach.....	9
1.3 ML for data stream classification.....	11
1.4 DL for data stream classification.....	14
1.4.1 Benefits of the DL approach.....	15
1.5 Feature selection.....	16
1.6 Research gap and direction.....	18
1.6.1 Problem definition.....	19
1.6.2 Research questions.....	20

1.6.3 Research objective	21
1.7 Research contributions.....	21
Chapter 2 Literature Review	25
2.1 Data mining and knowledge discovery.....	25
2.2 Data stream and significance of data stream classification.....	26
2.4 Deep learning and deep learning techniques for data stream classification.....	38
2.5 Different approaches for enhancing the efficiency of data stream classifiers.....	43
2.6 Recent techniques/classification models for data stream classification.....	48
2.7 Evaluation parameters for data stream classification.....	49
2.8 Conclusion	50
Chapter 3 The Deep Ensemble Algorithm Learning (DEAL) Framework.....	51
3.1 A novel DEAL framework for efficient classification of data streams.....	52
3.1.1 Learning phase	53
3.1.2 Optimization phase	61
3.1.3 Prediction phase	63
3.2 Conclusion	65
Chapter 4 DEAL and Extra Tree Ensemble Algorithm for Data Stream Classification	66
4.1 Algorithms for the proposed framework.....	67
4.1.1 DEAL Algorithm	67
4.1.2 ET Ensemble Algorithm	70
4.2 System requirements, Tools, and Libraries.....	73
4.3 Evaluation parameters.....	77

4.4 Conclusion	78
Chapter 5 DEAL Framework for Data Stream Classification	79
5.1 Implementation of DEAL framework for data stream classification.....	79
5.1.1 DEAL framework implementation on credit card dataset	80
5.1.2 DEAL framework implementation on stock prediction data set.....	83
5.1.3 DEAL framework implementation on hyperplane data set.....	86
5.1.4 DEAL framework implementation on sea generator data set	87
5.1.5 DEAL framework implementation on HAR data set.....	88
5.1.6 DEAL framework implementation on poker hand data set.....	91
5.1.8 DEAL framework implementation on RBF data set.....	93
5.2 Conclusion	95
Chapter 6 Performance Comparison of DEAL Framework and Statistical Analysis	96
6.1 Comparison with benchmark algorithms	96
6.2 Comparison with state-of-the-art algorithms	100
6.3 Statistical analysis for data stream classification.....	104
6.3.1 Statistical analysis for accuracy	105
6.3.2 Statistical analysis for categorical accuracy.....	117
6.4 Conclusion	130
Chapter 7 Conclusion and Future Research Directions.....	131
7.1 Summary.....	131
7.2 Contributions	133
7.3 Limitations and future directions.....	133
References.....	135

List of Abbreviations

S. No.	Abbreviation	Full form
1	AI	Artificial intelligence
2	ANN	Artificial neural network
3	ARF	Adaptive random forest
4	AUC	Area under curve
5	CCFD	Credit card fraud detection
6	CEP	Complex event processing
7	DA	Data analysis
8	DBMS	Database management system
9	DEAL	Deep ensemble algorithm learning
10	DL	Deep learning
11	DM	Data mining
12	DNN	Deep neural network
13	DT	Decision tree
14	DW	Data warehouse
15	EL	Ensemble learning
16	ET	Extra tree
17	FS	Feature selection
18	GI	Gini index
19	HAR	Human activity recognition
20	IG	Information gain
21	IoT	Internet of Things
22	KDD	Knowledge discovery in databases

23	KNN	k-nearest neighbor
24	LR	Linear regression
25	ML	Machine learning
26	MLP	Multiple layer perceptron
27	NB	Naive bayes
28	NN	Neural network
29	RF	Random forest
30	SVM	Support vector machine

List of Figures

Figure 1.1 Data analysis methods	2
Figure 1.2 Characteristics of data streams	4
Figure 1.3 Data stream mining methods	5
Figure 1.4 Classification process	6
Figure 1.5 Data stream classification approaches	8
Figure 1.6 Prediction process in ensemble approach.....	10
Figure 1.7 Steps in ML for classification.....	12
Figure 1.8 Workflows of the traditional ML process and DL process.....	15
Figure 1.9 Selection of relevant features.....	16
Figure 1.10 Solution set	20
Figure 1.11 Organization of the thesis	24
Figure 2.1 Knowledge discovery process	26
Figure 2.2 Data mining technologies for data stream classification	28
Figure 2.3 Data stream classification techniques.....	29
Figure 2.4 Challenges of ML techniques for data stream classification	37
Figure 2.5 DNN with three hidden layers	38
Figure 2.6 Optimization of deep learning model	42
Figure 2.7 Types of concept drifts in data streams	45
Figure 2.8 Ensemble feature selection	46
Figure 3.1 Phases of proposed framework.....	53
Figure 3.2 Steps involved in learning phase	54
Figure 3.3 Data preparation and visualization phase	54
Figure 3.4 Architecture of DL model.....	57
Figure 3.5 Layers of the DL model.....	59
Figure 3.6 Extra Tree Ensemble optimization	62
Figure 3.7 Prediction phase.....	63
Figure 3.8 Flowchart for the proposed framework	64
Figure 5.1 Normal and fraudulent transactions.....	80
Figure 5.2 Deal framework for CCFD.....	81
Figure 5.3 Comparison of prediction accuracy and categorical accuracy.....	82
Figure 5.4 Comparison of training and testing accuracy over CCFD dataset.....	83
Figure 5.5 DEAL framework for stock prediction.....	84
Figure 5.6 Comparison of prediction accuracy and categorical accuracy.....	85
Figure 5.7 Comparison of prediction accuracy and categorical accuracy.....	86

Figure 5.8 Comparison of prediction and categorical accuracy.....	87
Figure 5.9 Percentage of different activities	89
Figure 5.10 Comparison or prediction and categorical accuracy.....	90
Figure 5.11 Comparison of prediction and categorical accuracy.....	92
Figure 5.12 Comparison of categorical and prediction accuracy.....	93
Figure 5.13 Comparison of categorical and prediction accuracy.....	94
Figure 6.1 Performance comparison of DEAL with benchmark algorithms	99
Figure 6.2 Comparison of prediction and categorical accuracy.....	100
Figure 6.3 Performance comparison of DEAL with recent algorithms	102
Figure 6.4 Comparison of prediction and categorical accuracy.....	103

List of Tables

Table 2.1 Machine learning techniques for classifying credit card data streams.....	31
Table 2.2 ML techniques for classifying HAR data streams	32
Table 2.3 Machine learning techniques for classifying stock exchange data streams	34
Table 2.4 ML algorithms for data stream classification	35
Table 2.5 Deep learning techniques for data stream classification	40
Table 2.6 Ensemble approach for classifying data streams	44
Table 2.7 Ensemble-based feature selection algorithms for data streams.....	47
Table 2.8 Recent techniques/classification models for data stream classification.....	48
Table 2.9 Evaluation parameters.....	50
Table 3.1 Details of the nodes, layers, activation functions, and dropouts	60
Table 4.1 Notations for DEAL algorithm	68
Table 4.2 Notations for ET ensemble algorithm.....	71
Table 4.3 System details	74
Table 4.4 Tools and libraries	76
Table 4.5 Evaluation parameters.....	78
Table 5.1 Results of DEAL over credit card fraud detection data set.....	82
Table 5.2 Results of DEAL over stock prediction data set	85
Table 5.3 Results of DEAL over hyperplane data set.....	86
Table 5.4 Results of DEAL over SEA generator data set	87
Table 5.5 Activities in HAR data set	88
Table 5.6 Results of DEAL over HAR data set	90
Table 5.7 Results of DEAL over poker hand data set.....	91
Table 5.8 Results of DEAL over LED generator data set.....	93
Table 6.1 Implementation results of the DEAL framework and benchmark algorithms ..	97
Table 6.2 Average performance of DEAL and benchmark algorithms	99
Table 6.3 Implementation results of the DEAL framework and recent algorithms	101
Table 6.4 Average performance of DEAL and recent algorithms	102

List of Publications

Journal

1. Arya, M., & Sastry G, H. (2020). DEAL–‘Deep Ensemble ALgorithm’ Framework for Credit Card Fraud Detection in Real-Time Data Stream with Google TensorFlow. *Smart Science*, 8(2), 71-83. (Scopus Indexed, WOS) (Published)

DOI: <https://doi.org/10.1080/23080477.2020.1783491>

2. Stock Indices Price Prediction in Real-Time Data Stream using Deep Learning with Extra-Tree Ensemble (DELETE) Optimization

Journal: Journal of Computational Science and Engineering (Scopus Indexed, WOS) (in publication process)

DOI: 10.1504/IJCSE.2021.10040278

3. A Novel Extra Tree Ensemble Optimized Deep Learning Framework (ETEODL) for Early Detection of Diabetes” (Accepted)

Journal: Frontiers in Public Health, section Digital Public Health (SCI).

Conference

1. Arya, M., & Sastry G, H., “A novel deep ensemble learning framework for classifying imbalanced data stream”, in 2021 5th International Conference on Information and Communication Technology for Intelligent Systems, ICTIS 2021, IoT with Smart Systems, Volume 2.

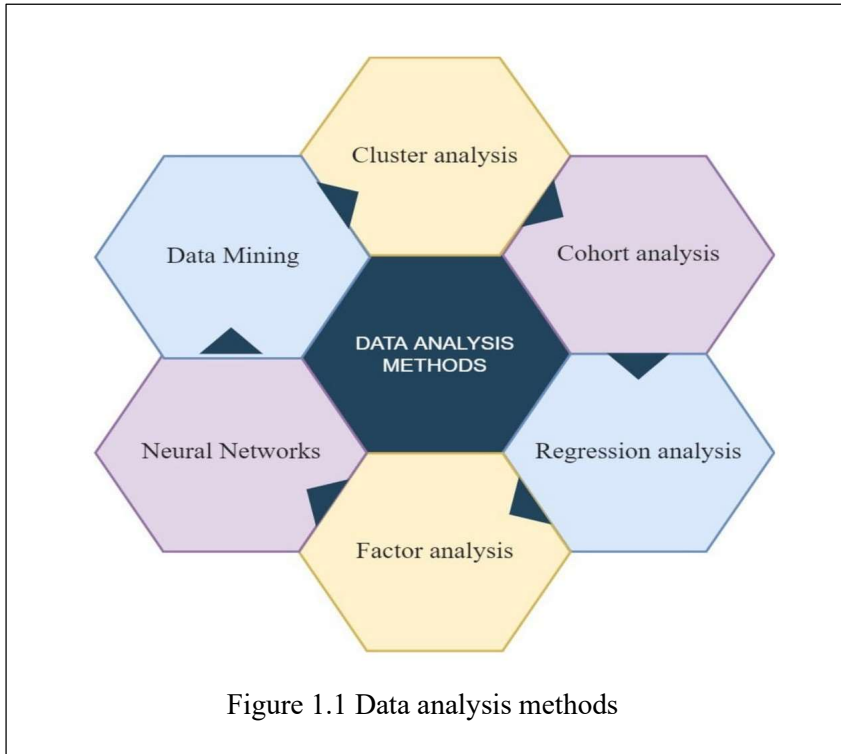
DOI: 10.1007/978-981-16-3945-6

Chapter 1

Introduction and Motivation

Data analysis (DA) has been a part of our lives since the dawn of human civilization. For example, the use of tally sticks to store and analyze data by early humans and the practice of carving grooves into sticks or bones by tribal people to forecast things like the duration of their food supplies show the data analysis techniques used in the past. In 1663, John Graunt conducted the earliest known experiment in statistical data analysis [1]. Before the invention of computers, it took the U.S. Census Bureau more than seven years to process the collected data and provide a final report in 1880. As a result, inventor Herman Hollerith created the “tabulating machine,” employed during the 1890 census [2]. The tabulating machine could process data from punch cards in a systematic manner. Thus, the 1890 census was completed in only 18 months.

DA is the systematic computational analysis of data or statistics [3]. It is a tool for finding, interpreting, and communicating basic patterns in data. It also implies using data trends to make more informed decisions. Businesses gain a significant competitive advantage by analyzing massive amounts of data using data analytics tools[4]. Data scientists mine information about a wide range of business activities—from current sales to historical inventory—and process that information in response to their queries using data analytics software. Various data insights are possible through data analytics tools like predictive analytics, business intelligence, and so on. Different types of DA methods are shown in Figure 1.1.



Data mining (DM) is the process of utilizing refined DA tools to uncover previously undiscovered valid patterns and relationships in massive data sets. These tools use statistical models, machine learning (ML) techniques, and mathematical algorithms such as neural networks (NN) or decision trees (DT). As a result, DM encompasses both analysis and prediction. DM dates back to the 1990s. Analyzing data in unconventional ways yields both surprising and valuable outcomes. In 2005 Roger Magoulas coined the term “Big Data” [5] to describe the tremendous amount of data that the business intelligence tools available then found nearly impossible to handle. In the modern world, sensors, IoT (Internet of things) devices, smartphones, and wearable health devices generate an enormous amount of data called the data stream, and data stream mining is the collection of meaningful information and knowledge from a data stream [6]. The most common methods of

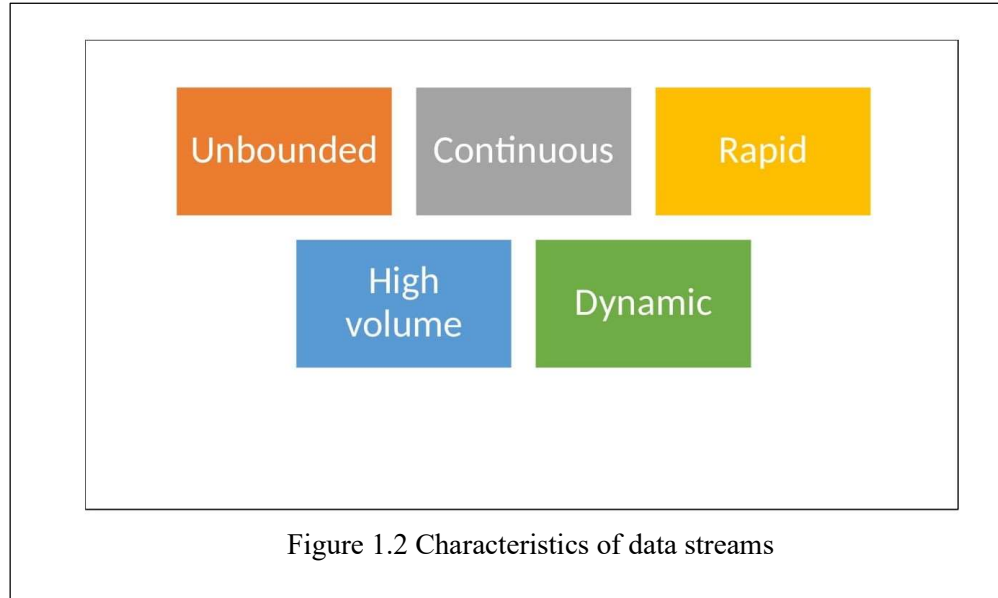
data stream mining are classification, regression, clustering, and frequent pattern mining.

1.1 Data streams and Data stream mining

A data stream is an ordered sequence of instances in time generated continuously by various heterogeneous resources. The data streams generated by varied resources have various formats and volumes. Some sources of data streams are as follows [7]:

- i. A wide variety of log files generated by customers using their mobile or web applications
- ii. E-commerce data
- iii. Gaming activity data
- iv. Information from social media sites like Twitter and Facebook
- v. Financial trading data
- vi. Geospatial services data
- vii. Credit card data
- viii. Stock exchange data
- ix. Weather forecast data

The data stream is used for different analytics, including correlations, aggregations, filtering, and sampling. Figure 1.2 shows the characteristics of data streams that make them different from static data.



Static data is operationally different from dynamic data in its method of storage and analysis. A static data environment stores the data prior to processing it. After processing and analyzing the stored data, the results are again stored in the database for further processing or used to make decisions. In contrast to this, in a dynamic data environment, data streams are processed as they are read, and the subsequent results are stored in a database. After the events are processed, they are discarded. These operational differences between static and dynamic data make it impossible to process a data stream using conventional DM techniques that can handle only static data. Because of this, data stream mining techniques are used to obtain insight from data streams.

Data stream mining is the process of extraction and discovery of knowledge and patterns from continuously generated data streams [6]. It is a research field that studies methods and algorithms to extract knowledge from volatile streaming data to utilize it in various domains. Data stream mining facilitates the analysis of an

enormous amount of stream data in real time using limited resources. Therefore, data stream mining algorithms should be incremental and respond quickly by processing each instance in a fixed amount of time and predicting at any time using limited memory. The most general approaches in data stream mining include classification, regression, and clustering. Figure 1.3 shows the categorization of data stream mining methods.

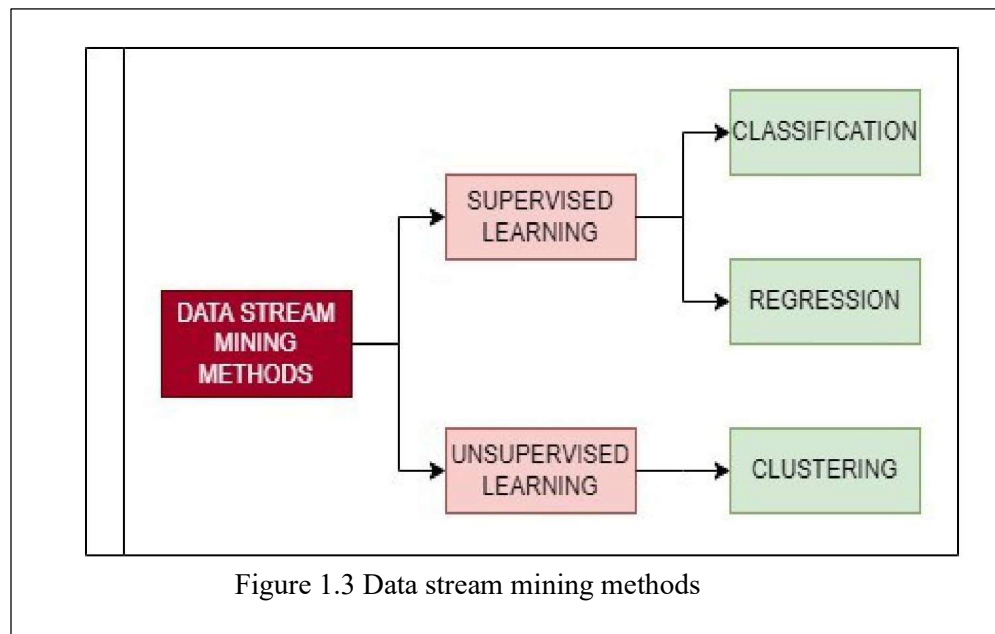


Figure 1.3 Data stream mining methods

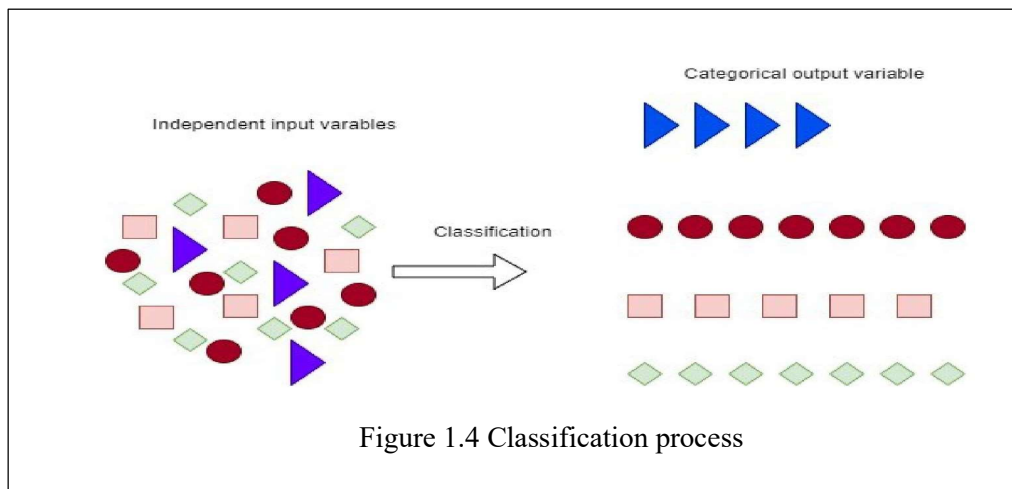
Clustering is an unsupervised learning technique for non-labelled data [8]. It is the process of making groups of a set of items such that the items in the same group are more similar than those in other groups. Clustering is applied during the description of data sets and as an initial step in various predictive learning tasks to better understand the information to be explored. Based on the clustering methods, there are several clustering algorithms [9]:

- i. k-mean clustering algorithm based on partitioning method DenStream
- ii. D-Stream and CEDAS algorithms based on density method

- iii. E-Stream and HUE-Stream based on the agglomerative method
- iv. CluDistream and SWEM based on the expectation-maximization method
- v. SNCStream based on social network data streams

Classification and regression are supervised learning techniques and need labeled data to train a model so that the trained model predicts the labels of unseen examples. A regression algorithm calculates the value of the target (response) as a function of the predictors. These correlations between predictors and targets are encapsulated in a model and subsequently applied to a new data set with unknown target values [10].

The classification task entails grouping data into categories, sometimes known as classes or labels [11]. Figure 1.4 shows the classification process.



The data to be classified can be either structured or unstructured. Unstructured data collects many different forms of data in their native formats, whereas structured data is very particular and stores data in a predefined format. Data warehouses (DW) store structured data, whereas data lakes store unstructured data

[12]. Classification is an important data mining technique that learns from historical data and predicts unknown instances. Earlier, classification techniques were used for static data. However, many applications that work on data streams have faced numerous challenges with traditional classification algorithms employed for dynamic data since the last decade. An ideal data stream classifier has the following properties:

- High accuracy
- Rapid adaptability to change
- Computational compatibility in comparison to traditional classification
- Scalable

Data stream classification is gaining popularity due to its use in various real-world applications, including e-commerce, banking, sensor data, and telephony records. It is a field of intense research nowadays, as there are opportunistic advantages for organizations. The organizations are able to make suitable business decisions based on knowledge extracted from data streams that can lead to a critical success accomplishment factor. Thus, data stream classification is attracting considerable critical attention [13].

The ideal data stream classification model is required:

- i. To be built around storing and processing only a limited amount of data because the data is generated and received potentially in an infinite sequence. Therefore, it is not practical to store it.
- ii. To be fast enough to handle high-velocity data so that each transaction is essentially processed in real time.

- iii. To be adaptive for changing concepts over time as data distribution may change over time, and the model built over data from the past might become pertinent for current predictions.

The features of data streams make the classification of data streams complex [14]. Due to these facts, there is a need to have an algorithm for efficient data stream classification and accurate prediction of unknown instances.

There are different approaches for classifying data streams. Per the application, the relevant approach is chosen to perform classification. Section 1.2 discusses the different approaches to data stream classification.

1.2 Different approaches for data stream classification

The most prevalent and well-studied technique for predictive DM and knowledge discovery is classification. Data stream classification is used for both offline and online streams [15]. According to the characterization, online stream categorization processes and updates the data as it arrives. This section discusses the classification approaches utilized in data stream classification. Figure 1.5 shows the various approaches used for data stream classification.

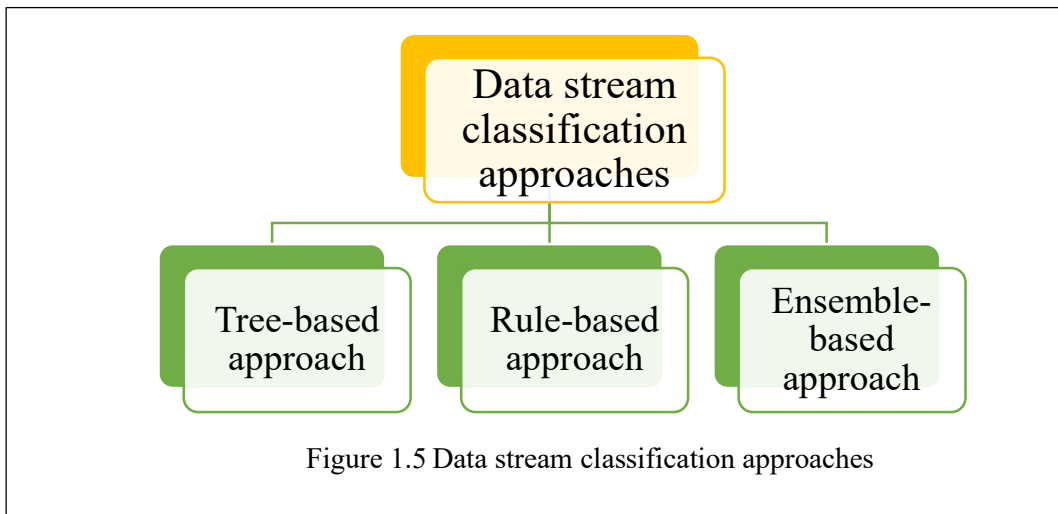


Figure 1.5 Data stream classification approaches

1.2.1 Tree-based approach

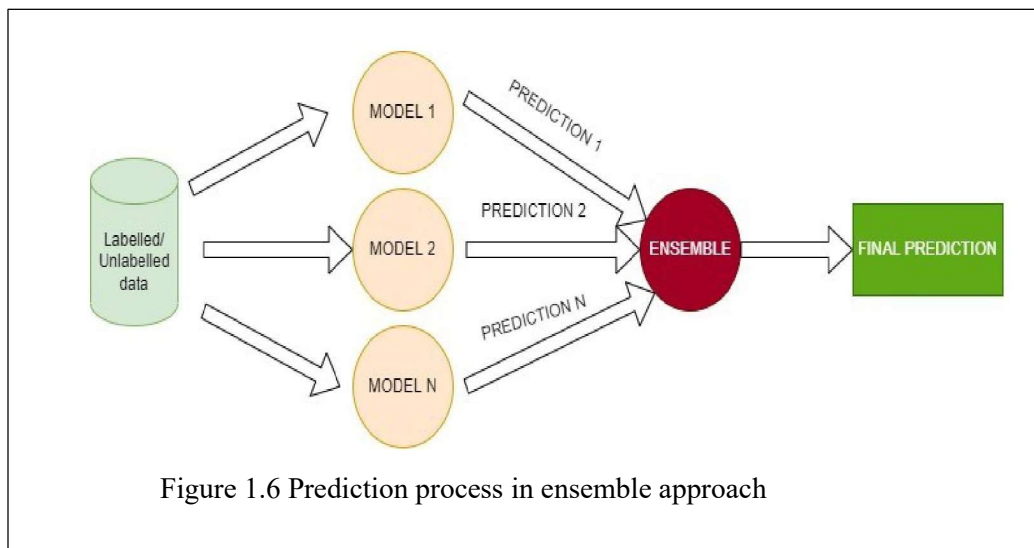
Classifiers using a tree-based approach are built around a decision tree. Tree-based classifiers produce predictions from one or more DT using a sequence of if-then rules. The nodes of the trees are the attributes. A tree grows by splitting the nodes. The user should specify a threshold value to control attribute splitting and maintain the best results for splitting below the threshold. A statistical analytical comparison is performed to delete fewer leaves and keep a count for all leave nodes in memory [9]. Tree-based classifiers are relatively fast to train and classify data that are not linearly separable. Hence, they achieve good performance. But they are somewhat unstable and difficult to generalize[16].

1.2.2 Rule-based approach

In a rule-based approach, classifiers use a set of rules for classification. The user in this classifier defines some criteria for generating rules, and these rules are generated during training [17]. Rule-based classifiers are frequently employed to create descriptive models. However, the downsides of the rule-based system are that they demand a great deal of manual work. This method also necessitates extensive topic expertise and is time consuming, as creating rules for a complicated system is challenging.

1.2.3 Ensemble-based approach

In Ensemble-based approach, instead of using a single classifier for classification, an ensemble of classifiers is used. An ensemble is a collection of multiple classifiers or a set of individual component classifiers whose outputs are pooled to predict the class of new incoming instances. For example, in the ensemble technique, individual predictions of several ML models are collected and grouped in some manner (e.g., by voting or weightage average) to form a final prediction. Figure 1.6 shows the predictions process in ensemble approach.



Most often, ensemble classifiers offer a more significant prediction performance and are far more scalable than individual classifier methods. Therefore, the ensemble learning technique is suitable [7] and widely used for data stream classification [18]. Furthermore, these techniques handle and control vast volumes of stream data and concept drifting features.

Among these discussed techniques to improve the performance of data stream classifiers, a well-established and challenging method is the ensemble approach. In the subsequent section, the significance of the ensemble approach in classifying data streams is discussed.

1.2.3.1 Significance of ensemble-based approach

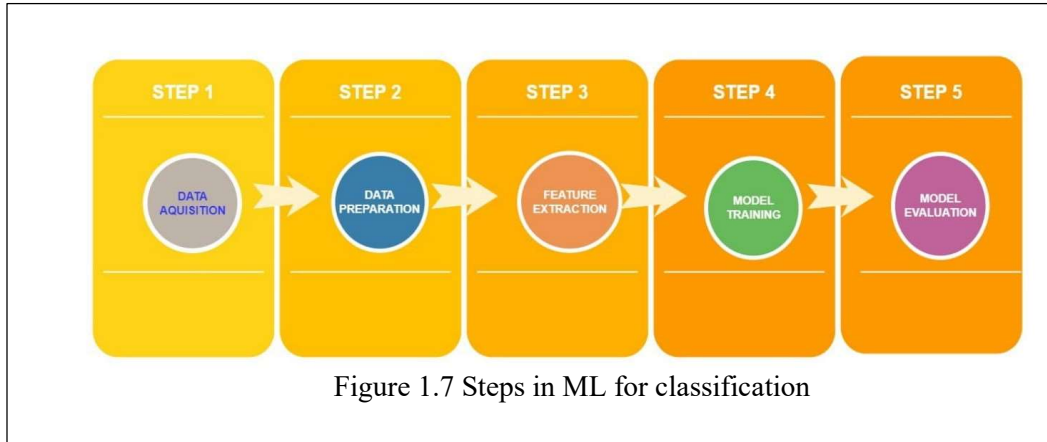
The ensemble-based approach plays a crucial role, particularly in dynamic environments like data stream learning [18]. They efficiently improve predictive accuracy and/or decompose a complex problem into more manageable sub

problems. Thus, this approach is suitable [19] and widely used for data stream classification. The ensemble-based classification makes it possible to extract useful information from dynamic data streams of various domains. For example, the text data from various social media platforms such as Facebook and Twitter are studied for extracting hot topics and expected trends. The ensemble-based approach can handle and control vast volumes of stream data and concept drifting features. Further, their integration with algorithms for detection of concept drift and incorporating dynamic updates, such as the selective removal or addition of classifiers, make them particularly useful. Due to its superior performance when compared to single learners, this approach has gained widespread acceptance due to the fact that it is relatively simple to implement in real-world applications for example numerical data captured by environmental sensor networks is studied to predict potential disasters like heavy rains, floods, storms, winds, or pollution peaks. In most DM applications, ensembles of classifiers are among the most effective classifiers.

Ensemble-based approaches for data stream classification are among the recent research areas in machine learning. Machine learning for data stream classification is discussed in Section 1.3.

1.3 ML for data stream classification

State-of-the-art ML algorithms perform learning, knowledge extraction, and visualization in data streams. Figure 1.7 shows the steps in ML for classification.



Much recent research specifies that an ensemble of ML classifiers significantly improves classification performance. Standard ML techniques try to build a model from the training data on one hypothesis, whereas Ensemble techniques try to construct a set of rules or hypotheses to use [20]. Classifiers comprising an ensemble are usually called base classifiers [19].

Ensemble learning(EL) is an ML paradigm where several learners train to resolve the same problem [21]. Using multiple classifiers of the same type or different types and combining those predictions improve predictive accuracy over just one model. Some well-known ensemble techniques are bagging, boosting, and stacking [22].

ML-based classification models attract researchers due to several reasons. First, models allow valuable pattern discovery in fast data streams, where transactions are generated as a continuous stream and each transaction is recognized by many parameters. Second, ML-based models or methods are good enough to discover existing patterns and strategies and identify new strategies coupled with unusual data behavior. Third, classification models based on ML methods are routinely integrated with novel feedback to achieve better accuracy. In Big Data analytics,

the challenging research area is building classification models for rapid data streams. The concept drifts in these high throughput data streams impose new challenges.

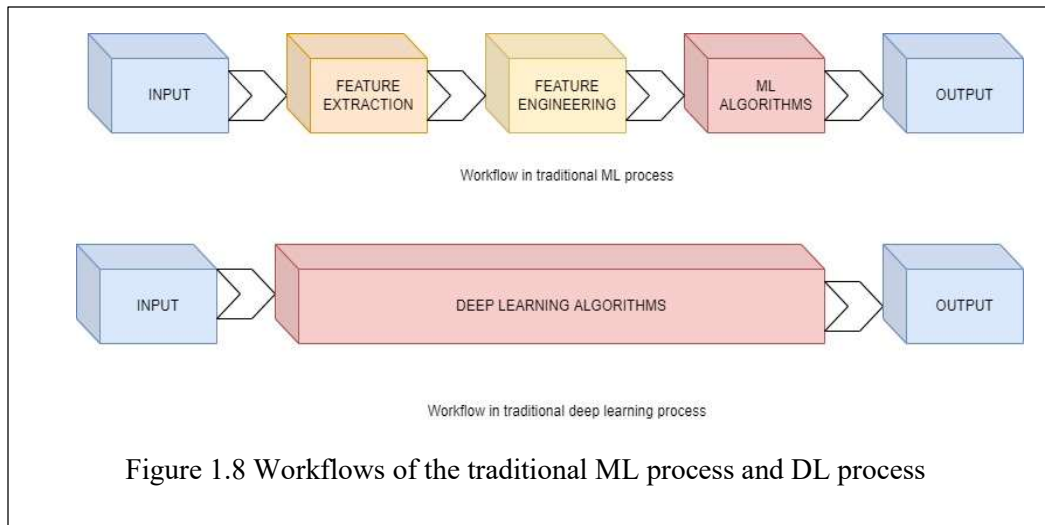
On the other hand, the expert system requires repeated feedback and rule revision, which is tedious and time consuming. Many applications exist in the domain, but they are either not parallel or have limited scalability. Developing effective and efficient data stream classifiers is challenging for the ML community because of the dynamic nature of data streams. Unlike conventional classification algorithms, data stream classification algorithms have to adapt with the change of data stream because of concept drift and feature evolution. However, conventional classification models remain stable once models are trained. ML is efficient, but it involves training static batch classifiers. But for regular data streams with time-varying characteristics, the model becomes outdated even before its deployment. Traditional classification methods assume that the discovered concept's statistical properties (that the model has predicted) are unchanged. Unfortunately, the occurrence of this phenomenon dramatically decreases classification accuracy. In traditional ML techniques, most of the applied features need to be identified by a domain expert to reduce the complexity of the data and make patterns more visible for learning algorithms to work.

Earlier studies also depict that ML algorithms need structured data for classification, have less prediction accuracy, are prone to overfitting, and require more computational time to predict. ML relies on manual extraction of relevant features and is thus bound by the domain knowledge of an individual. Traditional ML approaches extract low-level features that only recognize basic physical or postural activities and cannot recognize complex activities. Furthermore, ML does not exploit the temporal correlations between input samples. As a result,

unstructured data is hard to analyze for most ML algorithms. One approach to address these challenges and provide significant improvements over traditional ML approaches is deep learning (DL). DL models eliminate the need for using handcrafted and engineered feature sets in training. As a result, the models easily extract features that may not be obvious to the human eye. Furthermore, DL models boost accuracy [23]. The DL approach for data stream classification is discussed in detail in Section 1.4.

1.4 DL for data stream classification

DL models are more promising, as flexible DL networks make them suitable for both structured and unstructured data [20]. They are more capable of processing the data than the shallow neural network. Shallow neural networks have only one hidden layer as opposed to deep neural networks (DNN), which have several hidden layers, often of different types. DL analyzes structured as well as unstructured data from various sources. DL is rising in popularity due to its superiority in terms of accuracy when trained with massive amounts of data. The layered design of a DL approach allows it to learn categories in a gradual manner. It defines low-level categories such as letters in the initial layers. In the deeper layers, little higher-level categories like as words, and then higher-level categories such as sentences are defined. The performance of the DL model is further enhanced by optimization. Optimization minimizes error, cost, or loss (depending on the objective function) and provides more accurate results. The cost function is to be minimized because it describes the difference between the actual value of the estimated parameter and the model's prediction. The workflows of the traditional ML process and DL process are depicted in Figure 1.8.



DL has many benefits over traditional ML algorithms to classify data streams. The benefits of DL are discussed in the next section.

1.4.1 Benefits of the DL approach

The benefits of DL over traditional ML techniques have attracted the attention of researchers since the last decade. The benefits of the DL approach are as follows:

1. The DL approach is incremental in nature. When learning is done incrementally, DL accommodates new knowledge without retraining the existing model. The incremental learning nature makes it suitable for streaming scenarios.
2. DL identifies unknown classes during the testing phase and automatically updates itself if a new class is found. The adaptation with newly arriving classes makes DL beneficial in a data stream environment.
3. DL solves problems using an end-to-end learning technique. This technique enables DL to solve complex problems without having a deep knowledge of the problem domain. Therefore, DL learns in the complex environment of data streams.

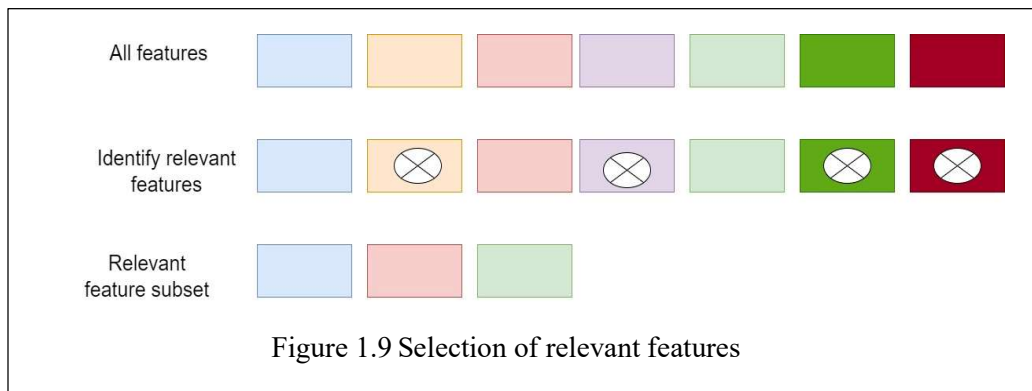
- DL learns high-level features and eliminates the feature extraction step. This makes learning faster in the data stream domain.

1.4.2 Limitations of DL models

Overfitting or underfitting is a significant limitation of DL models[24]. Underfitting occurs when a model fails to learn the problem adequately and performs poorly on a training data set while performing well on a holdout sample. Overfitting, on the other hand, occurs if a model learns the training data set too well, resulting in a model that performs well on the training data set but not on a holdout sample. Using feature selection (FS), it is possible to overcome the problem of overfitting or underfitting a model. FS is discussed in Section 1.5.

1.5 Feature selection

FS selects relevant features from the vast feature space, contributing to classification and discarding irrelevant features. It aims to select a subset of original features so that the feature space is optimally reduced based on the predetermined target. The goal of FS is to find the best set of features to build applicable models of studied phenomena. Figure 1.9 shows the selection of relevant features in FS.



FS has been successfully applied in classification problems in recent years. FS is one of the essential methods for influencing classification accuracy and improving algorithm predictive accuracy by reducing dimensionality, removing irrelevant features, and reducing the amount of data required for the learning process. FS reduces the time required for training the model, prevents overfitting, and improves the model's overall performance. The FS process seeks to reduce the data set dimension by analyzing and understanding the impact of its features. Their accuracy is predicted with the help of the classification model. The objective of FS is to eliminate redundant and irrelevant features. Usually, high dimensional data contains a high degree of irrelevant and redundant information. It may significantly degrade the performance of learning algorithms. Hence, it is always advisable to develop the induction algorithm for feature selection to increase the accuracy of the classifiers.

There are three types of FS methods: filter methods, wrapper methods, and embedded methods.

Filter methods: In filter method, features are filtered based on general characteristics (some metrics such as correlation) of the data set, such as correlation with the dependent variable. It is usually a faster and better approach for the data set with a large number of features. However, while it avoids overfitting, it may occasionally fail to select the best features. Correlation, chi-square test, and information gain are famous examples of filter methods[25].

Wrapper methods: Wrapper methods are based on greedy search algorithms, which evaluate all possible feature combinations and select the combination that produces the best result for a specific ML algorithm. Forward selection, backward elimination, and stepwise selection are typical examples of wrapper methods[26].

Embedded methods: The feature selection algorithm is integrated as part of the learning algorithm in embedded methods. Embedded methods combine the benefits of filter and wrapper methods[27]. The FS process is embedded in the learning or model building phases of the embedded method. Therefore, it is less computationally expensive than the wrapper method and less prone to overfitting. Regularization methods such as LASSO, ridge regression, and elastic net are examples of embedded method.

The present research program studies contemporary and relevant research work to identify the research gap and to formulate the problem statement. Chapter 2 presents the detailed literature survey. The literature survey identifies the research gaps for the present research program. Section 1.6 presents the research gap and the research direction in detail.

1.6 Research gap and direction

The appropriate methods and techniques remain an open challenge to deal with the classification of streaming data due to the following reasons:

- There is an unbalanced distribution of positive and negative classes in data.
- Conventional works overlook hidden transaction patterns.
- Most of the previous works have been evaluated on the conventional parameters only.
- Early models have also been criticized for overfitting—that is, models having less prediction accuracy than training or classification accuracy.
- The categorical accuracy is sometimes more important than overall accuracy, and in previous works it has not been considered in evaluation of the model.

The current research program aims to improve the efficiency of classification models for handling data streams and focuses on developing an efficient ensemble-based classification algorithm for data streams using the deep learning approach.

1.6.1 Problem definition

Efficient classification of streaming data is perceived as a multi-objective optimization problem. Newly arriving patterns, drifting concepts, and limited processing time are some of the constraints in developing an efficient classification model for data streams. Evaluation parameters like accuracy, F1-score, precision, and recall are not sufficient to define the efficiency of data stream classifiers.

Early models have been criticized for overfitting, the inability to find highly predictive features, and evaluating the model based on traditional evaluation metrics suited for static data environments. These evaluation metrics are not sufficient for data stream environments. In this research program, the aim is to develop and implement a framework for the efficient classification of data streams with the following solution set:

A solution set $S = \{s_1, s_2, s_3, s_4\}$ such that

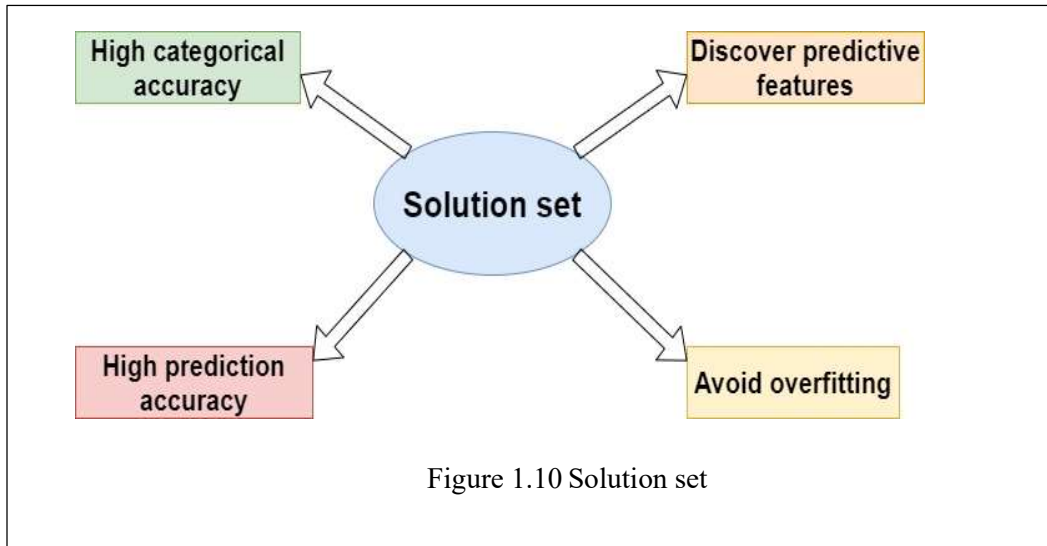
Where s_1 is categorical accuracy. It is to be increased.

s_2 is overfitting. It is to be decreased.

s_3 is model prediction accuracy. It is to be increased.

s_4 indicates predictive features to be discovered.

Figure 1.10 shows the solution set of the proposed framework.



1.6.2 Research questions

Several research questions emerge on the basis of the problem definition for data stream classification. The research questions are as follows:

- i. What can be the parameters for evaluating data stream classifiers?
- ii. How can the unbalanced data stream be handled?
- iii. How can overfitting/underfitting of data stream classifiers can prevented?
- iv. How to determine predictive features in a dynamic data stream environment?
- v. How can the overall efficiency of data stream classifiers be enhanced?

These research questions are the motivation for the present research program. Thus, a research objective was formulated based on these research questions.

1.6.3 Research objective

The objective of the present research program is to “Design and implement an efficient ensemble-based classification algorithm for real-time data streams in the context deep learning approach.”

Sub-objectives

- i. Studying the various existing ensemble-based classification approaches for real-time data streams.
- ii. Identifying the issues in existing ensemble algorithms of real-time data streams.
- iii. Exploring the various possibilities to improve a classifier’s efficiency to classify real-time data streams.
- iv. Devising an efficient algorithm for data stream classification using deep learning.
- v. Implementing the newly devised algorithm.
- vi. Performance testing of the newly implemented algorithm and comparing with existing algorithms to show the efficiency of the newly implemented algorithm.

1.7 Research contributions

The present research work focuses on “Designing and implementing an efficient ensemble-based classification algorithm for real-time data streams,” and its significant contributions are summarized as follows:

In this research work, an ensemble-based deep learning framework (DEAL) for classification of real-time data streams is proposed, and the effectiveness of the work is evaluated on the data sets taken from the UCI ML repository.

- i. The proposed framework is evaluated based on valuable metrics like categorical accuracy, prediction accuracy, F1-score, precision, and recall.
- ii. The extra-tree ensemble method is combined with DL in characteristic space in the DEAL framework, reducing the effect of highly unbalanced classes, thus yielding better performance.
- iii. The extra-tree ensemble optimization is used to minimize the cost function and improve the prediction accuracy. In addition, optimization is utilized to derive highly predictive and correlated features at the feature extraction step.

The current research emphasizes developing an efficient ensemble-based classification algorithm for the data stream using deep learning approach. Ensemble classifiers automatically adapt with the incoming drifts in the ensemble technique usage of multiple algorithms resulting in better predictive performance than using a unique algorithm.

The DEAL framework automates the feature extraction and optimally tunes features for the desired outcome. It extracts the distinguishing features. It is not required to extract features in advance, thus preventing time consumption in ML methods. An ensemble-based deep learning framework is proposed in this research to enhance the performance of real-time data stream classifiers while avoiding the overfitting problem. The robustness of the framework is achieved through an ensemble method and an optimization score function. The task of the optimization function is to minimize the loss. Feature reduction of the transaction is one of the primary challenges for ML classifiers. Therefore, a novel framework is designed that transforms the features of incoming transactions into tensors. These tensors are then supplied to fit into the model. Streaming data requires innovative processing techniques. Evaluating the model based on error and accuracy is imprecise. So, in

this work, a novel synthetic ensemble over different actual data samples is applied to optimize the DL model for better prediction. The proposed model performs better in the following aspects:

- i. Determines predictive features of streaming data
- ii. Improves prediction accuracy
- iii. Improves categorical prediction accuracy for unbalanced data sets
- iv. Prevents overfitting

The framework is compared with traditional ML techniques and recent works for various parameters like accuracy, F1-score, precision, recall, and computation time, and thus, it is concluded that the proposed framework's efficiency is better compared to conventional and recent works.

1.8 Organization of the Thesis

The Thesis is organized chapter wise so that the research work done can be read seamlessly. The Thesis consists of six chapters, starting with the introduction in Chapter 1, followed by literature review in Chapter 2, the proposed framework in Chapter 3, the algorithm of the proposed framework, the experimental setup and evaluation parameters in Chapter 4, implementation of DEAL framework for data stream classification in Chapter 5 and performance comparison and statistical analysis in chapter 6 and finally, conclusion and future direction in Chapter 7. The organization of the Thesis is shown in Figure 1.11.



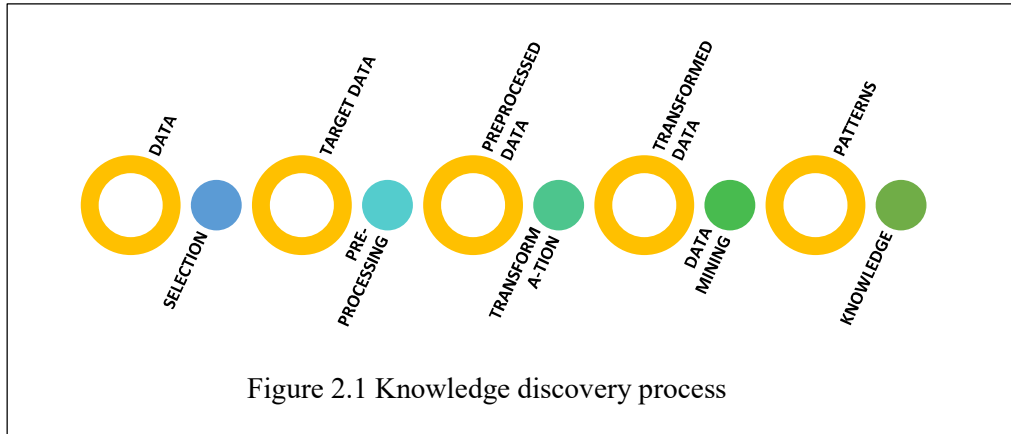
Chapter 2

Literature Review

The evolution of connected devices worldwide has resulted in the generation of massive amounts of data called the data stream. Data stream mining is the transformation of the data stream into valuable knowledge using data mining technologies. It is an emerging topic among researchers due to its significance in many real-life applications. Furthermore, the knowledge generated in mining data streams has a vital role in intelligent decision-making in different areas like economy, business, health care, and scientific research. With this context in mind, this chapter provides a systematic and detailed review of the significance of mining data streams followed by techniques for data stream classification. The goal of studying contemporary and relevant research work was to depict key challenges and issues in existing approaches for data stream classification. Research gaps were identified based on the challenges and issues in carrying out the current research program, on the basis of which, the problem statement was formulated. Further in this chapter, different approaches to enhance the efficiency of data stream classifiers like the ensemble-based approach and DL approach were studied to improve a classifier's performance. Finally, evaluation parameters for evaluating the performance of the data stream classifier were considered.

2.1 Data mining and knowledge discovery

DM refers to extracting common, previously unknown, and potentially useful information from a database. It is a part of the knowledge discovery process (KDD). Figure 2.1 shows DM as a step in an iterative knowledge discovery process.



In the decision-making process, information retrieval alone is not sufficient. DM is thus used to summarize data to extract useful information (i.e., for knowledge discovery and pattern identification in raw data) [28]. Classification, clustering, regression, association rules, outlier detection, sequential patterns, and prediction are widely used DM techniques. These techniques are used to extract information from static and dynamic data. Static data is a fixed data set—that is, data that does not change after it is collected. Dynamic data or data streams, on the other hand, change continuously. The present research focuses on the efficient classification technique for the data stream. Section 2.2 discusses the characteristics of data streams that make them different from static data and the significance of data streams classification.

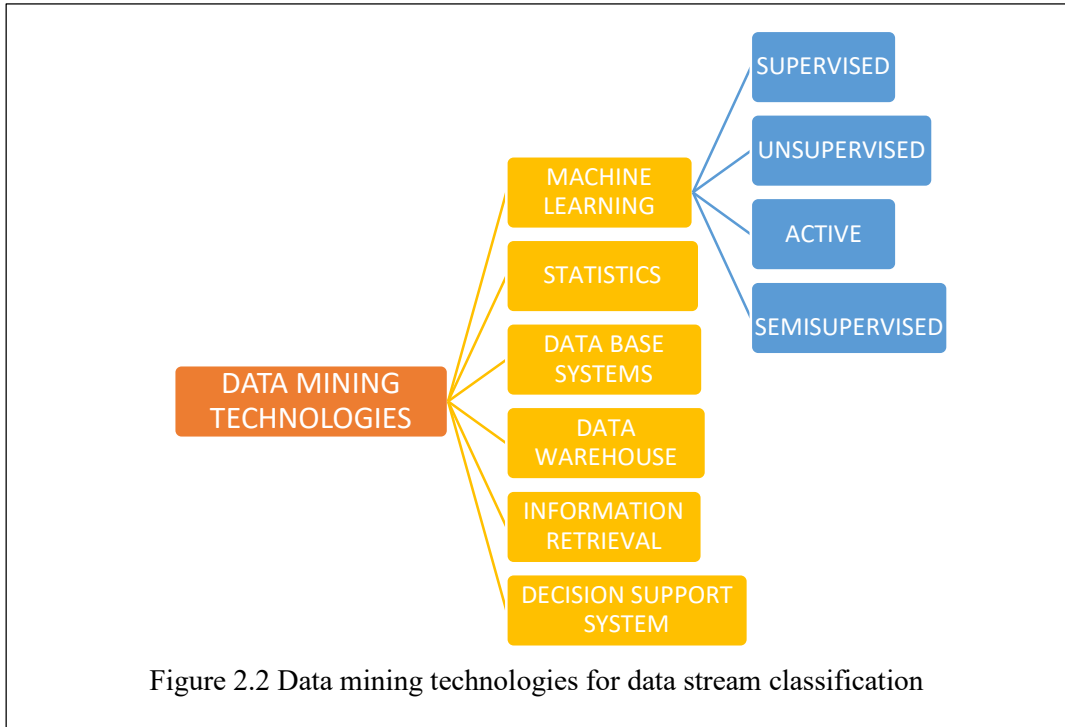
2.2 Data stream and significance of data stream classification

Data streams are high-volume, high-speed, and continuous data generated from various sources (e.g., IoT sensors, weblogs, smart devices, network traffic, health care devices, credit card transactions, multimedia data, scientific data) [29]. The data stream forms the primary source of Big Data [30]. Streaming data is dynamic and ever changing compared to traditional data, which is static [31].

Recently, data stream classification has attracted the attention of many researchers. It unmask a vast source of knowledge and has numerous real-life applications, including weather predictions, network monitoring, planning business strategies, credit card fraud detection, bio-surveillance, health monitoring, stock data analysis, and many more [32]. Data stream classification algorithms have high data processing requirements and need to perform well to meet real-time expectations. Some real-time applications of data stream classification are discussed here. Classification of credit card data stream to predict fraudulent and non-fraudulent transactions is one of the applications of data stream mining [33]. Another application is human activity recognition, or HAR for short. It is a vast topic of research that focuses on classifying and detecting a person's movement or action based on data streams obtained from sensors. Indoor positioning is seen as one of the open main applications of data stream classification. It is employed in several critical location-based services, such as indoor navigation at airports, hospitals, malls, warehouse tracking, and assisted living systems for aged care [34]. Predicting weather conditions worldwide is one of meteorologists' complex tasks and a challenging area in data stream classification [35].

The stock market generates a massive amount of data for the repository. Data stream classification algorithms are used to draw certain inferences from stock data analysis. One such inference could be the stock's trend [36].

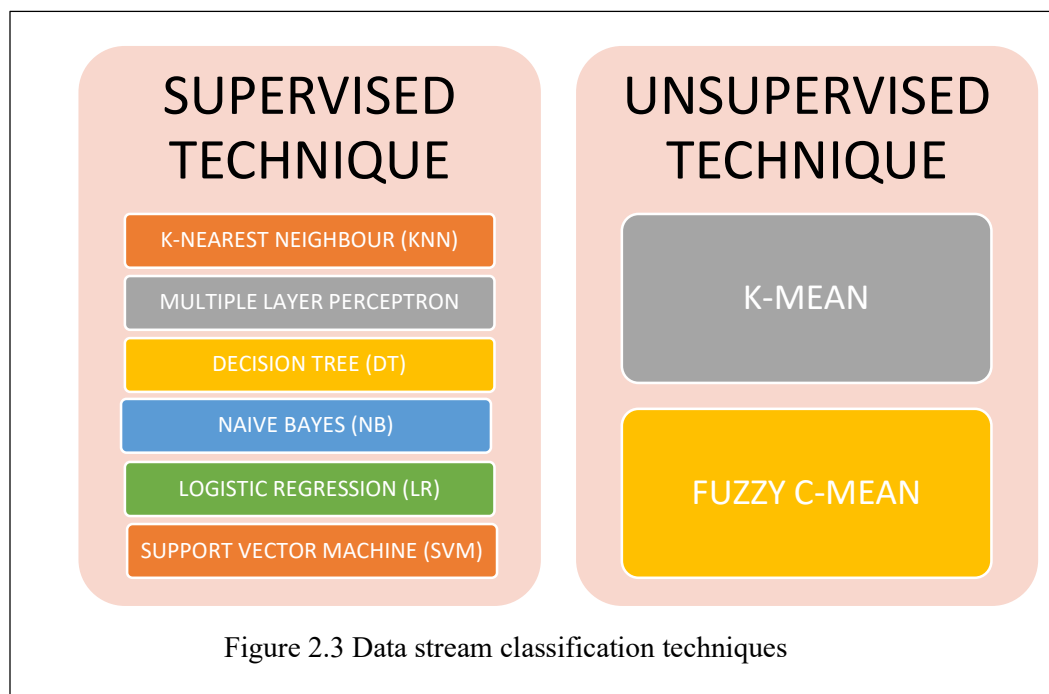
Data stream classification has gained increasing importance in several real-time applications; further enhancement in this area is becoming significantly prominent [37]. Several technologies are used to develop data stream classification methods. Figure 2.2 shows the data mining technologies for data stream classification.



Supervised ML is a sophisticated data stream classification technique for several reasons. First, ML is capable of analyzing enormous amounts of data and identifying specific trends and patterns that would otherwise go undetected by humans. Second, it has the ability to learn on its own, make predictions, and make improvements to algorithms. Third, as ML algorithms gather expertise, their accuracy and efficiency improve. Finally, ML algorithms are capable of dealing with multidimensional and multivarious data and data in a dynamic or uncertain state. Section 2.3 presents a survey of state-of-the-art ML techniques for data stream classification.

2.3 Machine learning and machine learning techniques for data stream classification

During the last couple of years, ML saw an exponential increase in its use and popularity, and this trend is expected to continue in the coming years. ML is a branch of artificial intelligence centered on the concept that systems can learn from data, recognize patterns, and make decisions with little to no human involvement. ML aims to draw out knowledge from data. It develops an automated computational model and continuously improves its performance based on experience. Unlike mining static data, ML algorithms need to adapt to the mannerism of data streams, which is dynamic, voluminous, and continuous. This section presents significant recent research in ML techniques for data stream classification. Figure 2.3 shows supervised and unsupervised ML classification techniques for data stream classification.



ML techniques for classifying data streams like credit card fraud detection (CCFD), stock prediction, and HAR are discussed here.

Obtaining information from the streaming data of credit card transactions is one of the most prevalent ways to commit fraud. Credit card information is illegally obtained and used to make fraudulent online purchases. It is hard for credit card companies and merchants to detect these fraudulent transactions amid thousands of routine transactions. In [38], the authors focused on the sampling technique to improve classifier performance while dealing with highly skewed data like credit card transaction data. They employed a hybrid undersampling and oversampling technique. They implemented several ML techniques like random forest, Naive Bayes, and multiple layer perceptron to detect fraud in the CCFD data set. In [39], the authors compared 10 ML algorithms, including the ensemble learning algorithm, over the credit card data set. They concluded that the ensemble learning algorithms performed better when the time feature was not in the data set. The authors in [40] compared the scalability of several ML techniques with the unbalanced data streams and concluded that the performance of the classifiers degraded when the data stream was unbalanced. The data-driven systems in the modern age necessitate the use of classifiers capable of dealing with unbalanced data streams. The evaluation of such data stream classifiers is still a challenge, as the existing evaluation measures are primarily concerned with accuracy. But accuracy is not enough to evaluate unbalanced data stream classifiers because the class ratio of unbalanced data stream changes over time. The authors in [41] proposed an incremental algorithm to calculate the area under the curve (AUC) metrics for evaluating data stream classifiers. The algorithm uses the combination of sorted tree structure and sliding window to compute AUC. Because of the high computing costs associated with AUC, it has not been successfully used for data stream mining until recently. In [42], authors focused their study on the fact that

better results were achieved over unbalanced or a skewed data set if it was preprocessed with the resampling (oversampling or undersampling) technique. They performed a comparative analysis of three ML algorithms: Naive Bayes, k-nearest neighbor, and logistic regression (LR). They concluded that improved performance could be achieved by applying sampling technique over an unbalanced data set before building the prediction model. In [43], authors employed random forest technique to classify credit card data. Table 2.1 summarizes the limitations and results of these ML approaches to classify credit card data streams.

Table 2.1 Machine learning techniques for classifying credit card data streams

ML technique	Limitation	Result	Data set
Multiple layer perceptron (MLP) [39]	Because MLP is fully connected, it has many drawbacks, such as having too many parameters.	Precision = 79.21% Recall = 81.63% Accuracy = 99.93%	CCFD
Naïve Bayes (NB) [43]	Naive Bayes is a supervised learning algorithm that ignores the interdependence of attributes. The Bayes theorem underpins it.	Precision = 95.9% F-measure = 84.65% Accuracy = 85.4%	
LR [43]	The assumption of linearity between the dependent and independent variables is a crucial constraint of logistic regression.	Precision = 95.1% F-measure = 91.3% Accuracy = 91.2%.	
k-nearest neighbor (KNN) [43]	KNN requires high memory to store all the training data, due to which it becomes computationally expensive.	Precision = 70.1% F-measure = 69.4% Accuracy = 67.9%	
Random forest algorithm [44]	The biggest drawback of random forest is that it becomes too slow and ineffective for real-time forecasts if there are too many trees.	Accuracy = 90%	

The classification of data streams obtained using sensors built into wearable devices, such as smartphones provides an opportunity for recognizing human activity. HAR helps identify human behavior and better understand an individual's health. HAR is an emerging research area due to its real-life applications in health care, the e-health system, and elder care for physically impaired people in a smart health care environment. In [44], the authors compared the performance of various ML algorithms over a HAR data set. In [45], the authors proposed a feature selection approach to select relevant features to improve the performance of state-of-the-art algorithms like KNN, MLP, and support vector machine (SVM). Table 2.2 summarizes the limitations and results of these ML approaches to classify HAR data streams.

Table 2.2 ML techniques for classifying HAR data streams

ML technique	Limitation	Result	Data set
SVM [45]	For big data sets, the SVM algorithm is ineffective. Moreover, when the data set contains more noise, such as overlapping target classes, SVM does not perform well.	Accuracy = 98.91 F-measure = 0.989 ROC area = 0.997 Kappa = 0.987	HAR data set
KNN [45]	The lazy learner is the name given to KNN (instance-based learning). During the training phase, it does not learn anything. The training data isn't used to derive any discriminative functions.	Accuracy = 97.66 F-measure = 0.977 ROC area = 0.986 Kappa = 0.9719	
CART [45]	The fact that it is a nonparametric technique is the most significant constraint; it is not advisable to generalize the underlying phenomenon based on the observed results.	Accuracy = 93.76 F-measure = 0.938 ROC area = 0.977 Kappa = 0.925	

They concluded that the feature selection approach makes the algorithms faster by reducing dimensionality. The authors in [46] used active learning algorithms to reduce classification time and passive learning algorithms to make the system accurate. The authors in [47] concluded that high memory consumption and the low value of the F1-score were the two challenging factors for HAR. In [48], the authors proposed a multivariate Gaussian framework to learn HAR features. The proposed framework showed improved performance over state-of-the-art algorithms.

Analysis of stock exchange data stream has gained popularity in the finance market. The analysis assists traders and investors in making buying and selling decisions, knowing the current price, and projecting future trends. The analysis is based on past and current data. In [49] and [50], the authors used various ML techniques to classify the stock exchange data set. The authors in [36] focused their attention on estimating sectors in a trend for a specific period. Sectors are the collection of stock data with similar characteristics. The sectors are estimated using feature extraction and clustering. The authors in [51] proposed a hybrid ML algorithm based on an artificial neural network (ANN) and DNN for forecasting daily return direction. They concluded that the accuracy of DNN-based classification had a higher accuracy than conventional ML algorithms. In [52], the authors employed the SVM technique to predict stock prices and concluded that SVM prevented the problem of overfitting. When it came to individual stocks, the authors of [26] concentrated their research on the use of network characteristic variables as input variables for KNN and SVM algorithms to predict the next-day fluctuation patterns of individual stocks. Table 2.3 summarizes the limitations and results of these ML approaches to classify stock exchange data streams.

Table 2.3 Machine learning techniques for classifying stock exchange data streams

ML technique	Limitation	Result	Data set
SVM [49]	For big data sets, the SVM algorithm is ineffective. Moreover, when the data set contains more noise, such as overlapping target classes, SVM does not perform well.	Accuracy = 0.8242 F-measure = 0.8438	Stock prediction data set
RF [49]	The biggest drawback of random forest is that it becomes too slow and ineffective for real-time forecasts if there are too many trees.	Accuracy = 0.9131 F-measure = 0.9178	
NB [49]	Naive Bayes is a supervised learning algorithm that ignores the interdependence of attributes. The Bayes theorem underpins it.	Accuracy = 0.8097 F-measure = 0.8193	
SVM with Softmax [50]	SVMs try to discover the maximum margin between data points of distinct classes, whereas the Softmax layer reduces cross-entropy or maximizes log-likelihood.	Accuracy = 57.22 F-measure = 63.16	

Apart from the state-of-the-art ML algorithms like SVM, Naive Bayes, KNN, and random forest (RF) for data stream classification, other approaches to enhance the overall performance of classifiers were also employed in the past. In [53], the authors proposed adaptive random forest (ARF) with resampling and adaptive operators to cope with the drifting concepts of data streams. They concluded that ARF was more accurate and used fewer resources than state-of-the-art algorithms. The authors in [54] concluded that the generalization capability and accuracy of data stream classifiers could be improved by adding more hidden layers in extreme learning machines. Table 2.4 summarizes various ML algorithms and other similar works like Active learning classifier [55], Similarity-based data stream classifier (SimC) [56], CVFDT [57], dsCART [58], FS-SVM [59], Hoeffding adaptive tree (HAT) [60] and IDS-ELM [61]

for data stream classification along with the technique, evaluation parameters, and type of data set used.

Table 2.4 ML algorithms for data stream classification

Algorithm	Technique/methodology	Evaluation parameters					Data set	
		Accuracy	Kappa M	Kappa T	Error rate	Updation time	Synthetic	Real time
ARF [45]	ARF is included with resampling and adaptive operators to cope with the changing concepts of data streams.	✓	✓	✓			✓	✓
DELM [46]	Online learning mechanism is used to train extreme learning machines (ELM) to improve their performance.	✓					✓	
Active learning classifier[47]	The catastrophic forgetting mechanism is used to update the classifier by forgetting outdated concepts.	✓					✓	
Similarity-based data stream classifier (SimC) [48]	It employs instance-based learning techniques for classifying data streams. Instance-based learning allows instant removal of outdated concepts and insertion of newly arrived concepts to maintain efficient performance of the classifier.	✓				✓	✓	
CVFDT [49]	CVFDT is based on a very fast version of VFDT. CVFDT learns from concept drift and updates the decision tree with incoming data streams.	✓					✓	
dsCART [50]	This algorithm is a modification of VFDT. It is used to determine whether the best attribute to split the considered node obtained from a finite data sample is also the best attribute to split the entire data stream.	✓					✓	✓
FS-SVM [51]	FS-SVM is based on the support vector machine and feature selection method.				✓			✓

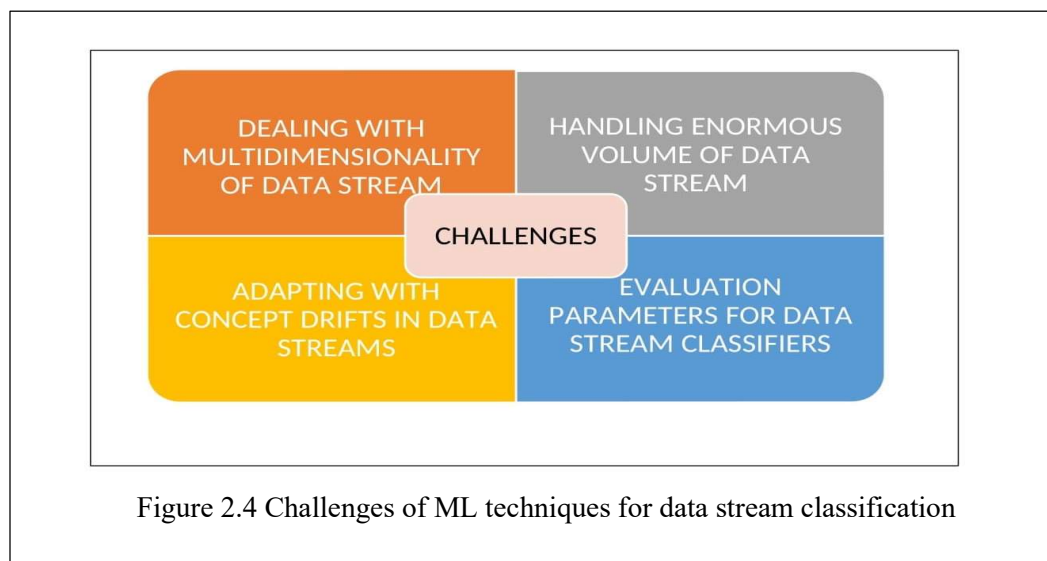
Table 2.4 continues.....

Algorithm	Technique/methodology	Evaluation parameters					Data set	
		Accuracy	Kappa M	Kappa T	Error rate	Update time	Synthetic	Real time
Hoeffding adaptive tree (HAT) [52]	Hoeffding adaptive tree is adaptive, as it learns from data streams. The main advantage of HAT is that it does not need a sliding window of fixed size.				✓			✓
IDS-ELM [53]	IDS-ELM uses a search mechanism to reduce the number of neurons in the hidden layers to maintain the performance of the classifiers.	✓					✓	✓

ML plays a significant role in DM. ML algorithms discover hidden anomalies and exceptional patterns. Surveying the past and recent related studies reveals that classifying data streams using traditional ML approaches has various challenges and issues due to the enormous volume, high speed, veracity, and changing concepts of data streams. The first challenge is handling the enormous volume of the data stream. Mining such a considerable volume of data requires unbounded memory, while the available memory for processing is limited.

Therefore, mining data stream demands high performance in terms of accuracy and speed. Also, the parameters have to be modified dynamically during the interactive mining of data streams [62]. The second challenge is the adaptation of data stream algorithms with the changing concepts of data stream known as concept drift. The algorithms for the data stream need to be incremental to address the concept drifts occurring in data streams. The third challenge is the evaluation of data stream algorithms. Data stream algorithms require novel evaluation methods and metrics, as the traditional metrics are designed for algorithms using static data

[63]. However, data streams are dynamic and change with time. Thus, historical data becomes irrelevant for mining. The fourth challenge is handling the multidimensionality of the data stream, which needs sophisticated mining techniques. The fifth challenge is avoiding overfitting or underfitting data stream models [64]. These challenges and issues need to be resolved for developing efficient classification algorithms for data streams. Figure 2.4 shows the challenges faced by traditional ML techniques in data stream classification.

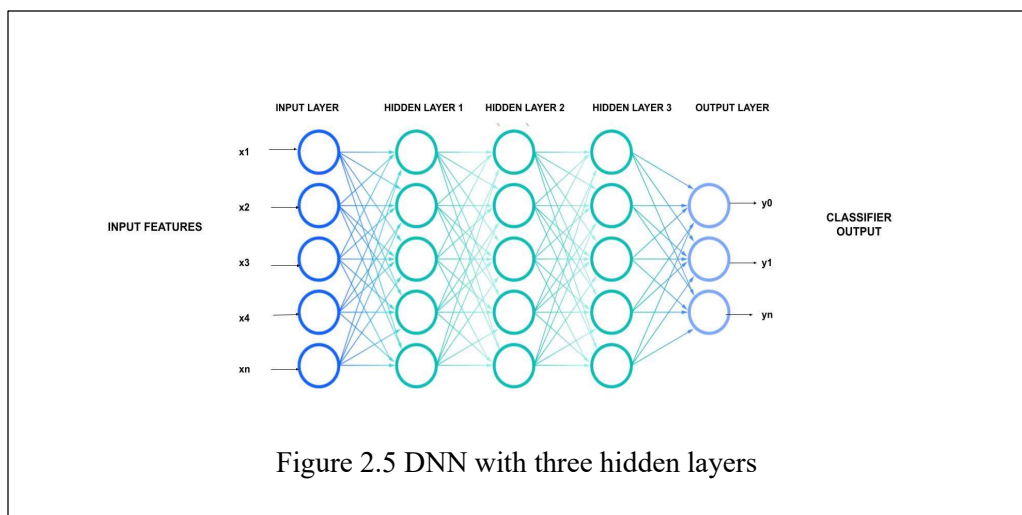


Related studies for addressing the above-mentioned challenges diverted the attention of the researchers towards DL approach. DL is one approach that can address these issues while also providing significant improvements over traditional machine learning approaches. DL models eliminate the need to use handcrafted and engineered features in training. As a result, the models easily extract features that may not be obvious to the human eye. Furthermore, DL models boost the accuracy of the classifier. Section 2.4 discusses the DL approach for data stream classification in detail.

2.4 Deep learning and deep learning techniques for data stream classification

A DNN is an ANN that has multiple layers between the input and output layers. DL is a ML and artificial intelligence (AI) technique that is designed to mimic the way humans learn.

DL is used extensively in data science. DL is beneficial for data scientists responsible for gathering, analyzing, and interpreting massive amounts of data; it speeds up and simplifies the process. It is a means to automate predictive analytics at its most basic level. DL algorithms are built in a hierarchy of increasing complexity and abstraction unlike typical ML algorithms, which are linear. A domain expert identifies most of the valuable features in traditional ML approaches to minimize data complexity and make patterns more evident for learning algorithms to work. The most significant benefit of DL algorithms over ML algorithms is that they attempt to learn high-level features from data incrementally, reducing the need for domain expertise and the extraction of hard-core features [65]. DL is a layered structure, and it has many hidden layers between the input and the output layers. As a result, such networks are referred to as “deep.” For example, Figure 2.5 shows a DNN with three hidden layers.



DL is learning multiple levels of representations and abstractions. These multiple layers help to extract knowledge and information from Big Data. DL can utilize unlabeled data during training, solve complex problems that require discovering hidden patterns in the data, combine them, build much more efficient decision rules, and understand relationships between a large number of interdependent variables.

The most significant distinction between DL and regular ML is its performance when data scales up. Traditional methods cannot deal with Big Data, as they cannot extract nonlinear patterns and solve Big Data area problems. Knowledge has always been the key to success.

Furthermore, it makes machines independent from humans by extracting helpful information from unsupervised data without human intervention. Thus, DL models are better than ML models and are more efficient because they need less computation [66]. Table 2.5 lists the DL techniques like Adaptive incremental deep learning solver [67], Streaming deep forest (SDF) algorithm [68], GRU-FCN [69], Stacked sparse autoencoder (SSAE) [70], Asynchronous dual-pipeline deep learning algorithm [71], Hybrid deep learning model [72], GUS-DL algorithm [73], and Deep super learner [74] for data stream classification along with the description, data set, and evaluation parameters.

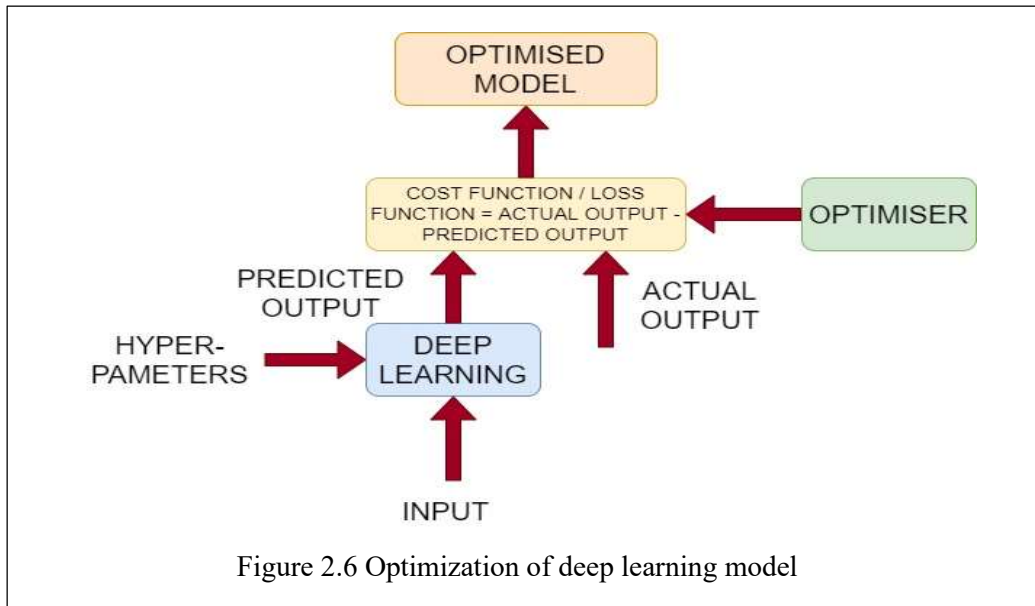
Table 2.5 Deep learning techniques for data stream classification

Algorithm	Description	Evaluation parameters	Data set
Adaptive incremental deep learning solver [67]	The degree of concept drift and the regression model's training curve dynamically set the batch size and epoch for the given temporary mini-batch.	RMSE, average training time	AMI data set
Streaming deep forest (SDF) algorithm [68]	It's a stream-specific version of the gcForest model. By reusing gcForest's cascade structure, SDF keeps gcForest's representative learning capabilities. In addition, the standard random forest at each layer is replaced by ARF, a high-performance forest model for the stream scenario, to update the model on the fly.	Accuracy	11 synthetic data streams and 9 real-world data sets
GRU-FCN [69]	The temporal FCN extracts feature from sensor-source data sets. At the same time, the GRU allows the model to distinguish temporal dependencies within these sequential data streams, allowing the model to learn both features and temporal dependencies to predict the proper class for each sensor data set.	Classification error, mean per class classification error (MPCE), and arithmetic average rank	18 different sensor source-obtained data sets
Stacked sparse autoencoder (SSAE) [70]	It learns a high-level feature representation of raw data unsupervised. These high-level properties allow the classifier to classify shows in TV streams quickly.	Classification rate	TV streams
Asynchronous dual-pipeline deep learning algorithm [71]	The proposed system features two independent layers for training and testing that work simultaneously to deliver speedy predictions and make frequent model updates.	Processing time, accuracy	29 different time-series data sets

Table 2.5 continues....

Algorithm	Description	Evaluation parameters	Data set
Hybrid deep learning model [72]	It enhances GWO's capabilities by combining grey wolf optimization (GWO) and convolutional neural network (CNN).	Detection rate, false positives, and accuracy	Streaming network traffic data
GUS-DL algorithm [73]	E-commerce employs a variety of solutions to strike a balance between prediction, accuracy, precision, specificity, sensitivity, and usefulness of data.	Prediction, accuracy, precision, specificity, sensitivity	E-commerce data set
Deep super learner [74]	The super learner is a method that uses a variety of methods to choose the best weights to use when generating the final prediction. The super learner ensemble is extended by DSL.	LogLoss, accuracy	Image and text data set

DL is a process that requires many iterations. Experiments with various permutations are done to discover which combination of hyper-parameters works best. As a result, it's critical for the DL model to train in less time without sacrificing quality. The goal of DL is to minimize the difference between actual and predicted output. The difference is known as cost function or loss function. Figure 2.6 shows the optimization of the DL model.



Optimizers are techniques or approaches that adjust the characteristics of the DNN, such as weights and learning rate, to reduce cost function or loss function. The optimizers determine how to alter the neural network’s weights or learning rates to reduce losses. Optimization algorithms or methods are in charge of lowering losses and delivering the most accurate results. Many researchers experiment with different techniques to optimize the DNN to make more accurate predictions. For example, the authors in [75] proposed that feature selection optimized the DNN and improved prediction accuracy. They experimented with 11 feature selection algorithms and suggested that feature selection optimized the DNN in many cases. In [76], authors commented that feature selection optimization could generalize unknown data, avoid overfitting, boost the average accuracy of the classifier for both two-class and multiclass data sets and lower the cost of measuring feature values. Section 2.5 discusses the different approaches to increase the efficiency of the data stream classifiers based on the DNN.

2.5 Different approaches for enhancing the efficiency of data stream classifiers

The utilization of intelligent IoT devices, sensors, and social networks has resulted in an enormous volume of IoT data streams generated every day from several applications that can be turned into valuable information using ML tasks. Several crucial difficulties occur in practice in extracting valuable knowledge from these dynamic data streams; the most important is that the stream must be efficiently handled and processed. In the literature, several works aimed to enhance the efficiency of data stream classifiers. In [77], the authors suggested a summarization technique for data stream analytics. In that technique, a synopsis of the information gathered from stream instances was preserved by either storing a tiny portion of the incoming data or creating alternative data structures that stored a summary of the data. Some techniques of summarization are sampling, histogram, sketching, and dimensionality reduction. In some past works [18] [78], researchers discussed combining multiple models to improve the accuracy of the final ensemble model. The outputs of the individual models were aggregated to reduce model error and maintain generalization.

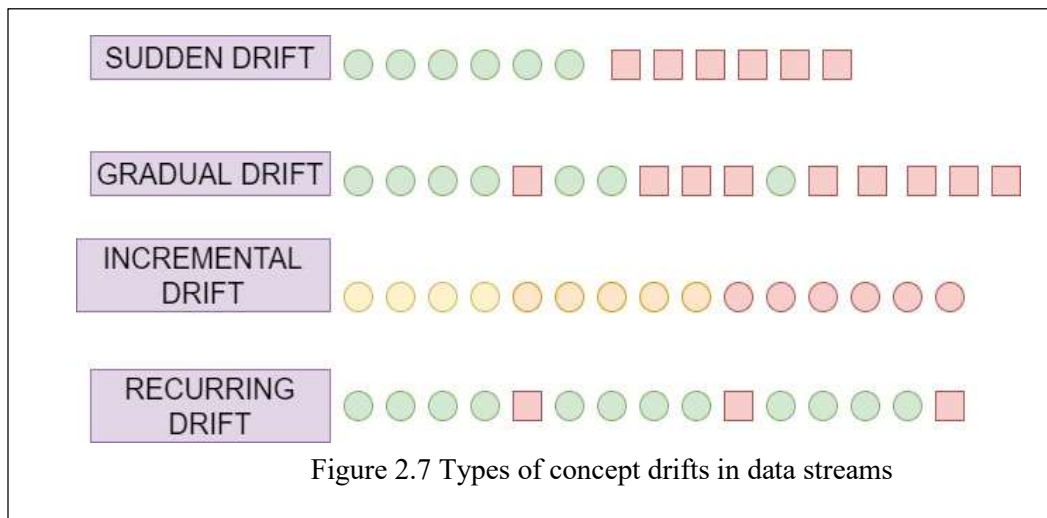
Other techniques like creating ensemble models for data stream classification have also attracted the attention of many researchers [19]. The ensemble approach combines multiple models known as base models to reach the final decision. The base models differ from one another. Therefore, by ensembling the model's decisions, the accuracy of the result increased significantly. This fact has increased the popularity of the ensemble-based approach in machine learning among researchers; Table 2.6 lists the popular ensemble-based approaches like Ensemble for nonstationary data stream (ENSDS) [79], CWEOS-ELM [80], OOB and UOB ensemble learning [81], DDCS [82], MSRS [83], and SAE [84] for classifying data streams.

Table 2.6 Ensemble approach for classifying data streams

Algorithm	Description	Evaluation parameter	Data set used
Ensemble for nonstationary data stream (ENSDS) [79]	The ensemble of classifiers in ENSDS is built by classifiers generated using current data and combined using majority voting.	Recall, F-measure, G-mean	Synthetic data set
CWEOS-ELM [80]	The algorithm provides faster learning by employing a self-adapted weight mechanism.	Prediction accuracy	Synthetic and real-time data sets
OOB and UOB ensemble learning [81]	The resampling method is used with the original OOB and UOB ensemble to obtain improved performance.	Accuracy, G-mean	Synthetic and real-time data sets
DDCS [82]	The application of dynamic classifier selection in data stream classification is described here. DDCS constructs an ensemble by treating the stream as a collection of data chunks and adding new classifiers as the data chunks arrive in the ensemble.	Processing time and memory consumption	Synthetic and real-time data sets
MSRS [83]	The ensemble is based on a random subspace sampling approach and data balancing to ensure diversity among base classifiers.	Balanced accuracy	Synthetic and real-time data sets
SAE [84]	In SAE, the relationship information between base classifiers forms a subnetwork of similar classifiers. These subnetworks are ensemble to form a network.	Accuracy, processing time	Synthetic data set

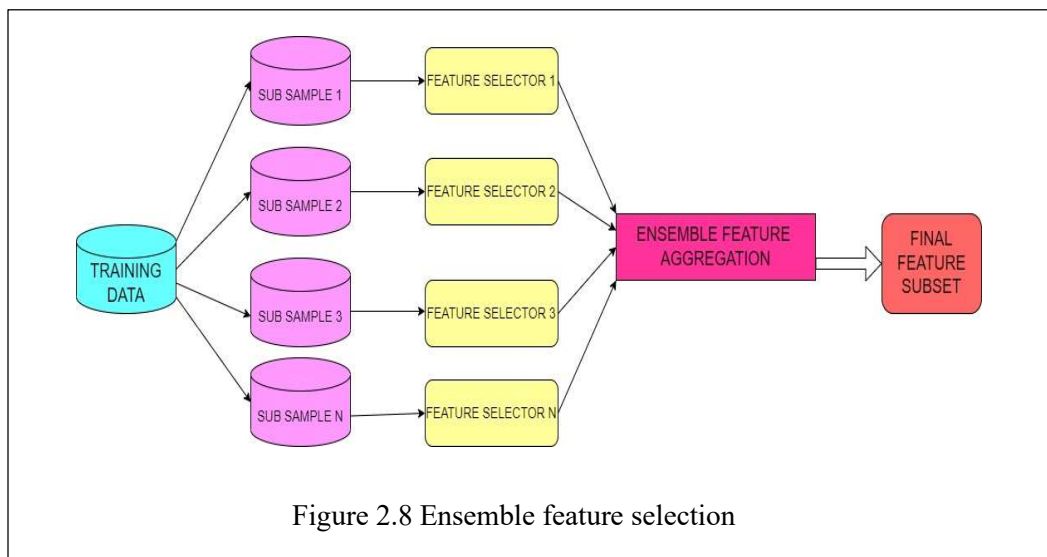
Furthermore, the authors in [85] suggested that using an appropriate preprocessing approach positively impacted the classification outcome. Therefore,

the authors applied complex event processing (CEP) to define preprocessing rules for data streams. The efficiency and performance of the data stream classification algorithm were also affected by the dynamic nature, high dimensionality, and the concept drifting nature of data streams. Concept drift in data streams occurs when the old features become obsolete and new features emerge with time. Drifts or changes in data streams can be sudden, gradual, incremental, or recurring. Figure 2.7 shows the types of concept drifts in data streams.



Therefore, it is challenging for researchers to design an efficient algorithm for classifying data streams. Selecting relevant features from the available features and discarding irrelevant ones is called feature selection. It is one of the techniques to improve the efficiency of data stream classifiers because in data streams, the relevant subset of features changes over time. The ensemble concept is applied to FS tasks to get the most relevant and recent optimal subset of features. Ensemble FS produces a more robust result for classifier learning tasks and handling concept drifting features. In [20], the authors applied an ensemble-based strategy that combined multiple feature selection procedures to provide an

aggregate result that reduced the variance of a single result and eliminated irrelevant features. The authors in [86] combined FS techniques such as filtering and wrapping. The filtered phase calculated and scored each feature’s information gain (IG) before passing it on to the wrapper phase. The wrapper phase looked for the best subset of features toward the end. Figure 2.8 demonstrates the process of ensembling different feature selection methods to obtain the final feature subset.



In [87], the authors proposed a method that used the Gini index (GI), IG, SVM, and LR to develop a two-level ensemble for FS. The authors in [88] extended mRMR to mRMRe—an ensemble variation that builds and ensembles numerous feature sets rather than a single list of features. An ensemble-based wrapper for feature selection from data with a highly skewed class distribution was used in [89]. The main idea was to use sampling to build several balanced data sets from the original unbalanced data set and then use an ensemble of base classifiers trained on each balanced data set to evaluate feature subsets. In [90], the authors used an extra ensemble tree for feature selection. The extra tree ensemble feature selection method used an ensemble of extra tree classifiers to determine the essential

characteristics that contributed to the class target. A hybrid feature selection technique was used in [91]. It combined three feature selection approaches for selecting significant features: chi-square, IG, and principal component analysis (PCA) to create a unique hybrid feature selection strategy that performed well across all data sets. In FSE algorithm [92] also, authors used ensemble-base feature selection for data streams . The techniques discussed here are used for either binary classification or for multiclass classification. Table 2.7 gives the description of the data set and the category of classification for which the technique is used.

Table 2.7 Ensemble-based feature selection algorithms for data streams

Algorithm	Dataset used	Category of classification	
		Binary	Multiclass
Ensemble feature selection [86]	Cyberattack data set	✓	
A novel feature selection ensemble approach [87]	Text data set for sentiment analysis	✓	
Extended minimum redundancy maximum relevance (mRMRe) [88]	Large data sets of cancer cell lines		✓
An ensemble-based wrapper approach for feature selection [89]	Iris, blood cancer, pima data set		✓
Extra tree ensemble-based feature selection [90]	Android apps		✓
Hybrid feature selection [91]	Skin disease data set	✓	
FSE [92]	Number of data sets like arrhythmia, Cleveland, glass, heart, ionosphere, Libras, oligos, ozone, secom, sonar, water, and wine		✓

In the present research work, the extra tree ensemble-based feature selection technique was employed to optimize classification results to enhance the efficiency of the data stream classifier.

2.6 Recent techniques/classification models for data stream classification

Recently, many researchers have been involved in finding the techniques for improving the efficiency of data stream classifiers. Table 2.8 summarizes some of the recent advancements in this area like DISSFCM (Dynamic incremental semi-supervised FCM) [93], Similarity-based data stream classifier (SimC) [56], LELC [94], PAW [95], Self-paced ensemble algorithm [96], An iterative boosting-based ensemble algorithm [97] and A clustering and ensemble-based classifier algorithm [98] along with the possible limitations and evaluation parameters used to evaluate the model/technique/algorithm.

Table 2.8 Recent techniques/classification models for data stream classification

Algorithm	Description	Evaluation parameters	Limitations
DISSFCM (Dynamic incremental semi-supervised FCM) [93]	It is based on an incremental semi-supervised fuzzy clustering algorithm. The method assumes that partially labeled data belongs to different classes continuously available during time in chunks. Each chunk is processed by semi-supervised fuzzy clustering leading to a cluster-based classification model.	Accuracy, recall, precision, F1-measure	It is limited to the classification of data with homogeneous class distribution.
Similarity-based data stream classifier (SimC) [56]	It improves performance by introducing a novel insertion/removal policy, which adapts quickly to the data tendency and maintains a small representative set of examples and estimators that guarantee reasonable classification rates.	Mean classification time, mean updating time, accuracy	It cannot deal with abrupt concept changes.
LELC [94]	It requires a small set of positive examples and unlabeled examples that are easily obtainable in the data stream environment to build accurate classifiers.	Accuracy, speed	Application limited to text data set.
PAW [95]	It improves the windowing technique with a mechanism to include older examples as well as the most recent ones, thus maintaining information on past concept drifts while being	Accuracy, RAM hours	It is tested for limited performance metrics.

Table 2.8 continues..

Algorithm	Description	Evaluation parameters	Limitations
Self-paced ensemble algorithm [96]	The algorithm is based on the classification hardness. In this algorithm, instead of simply balancing the positive/negative data or directly assigning instance weights, classification hardness is distributed over the data set. Then, iteratively select the most informative majority data samples according to the hardness distribution.	F1-score, G-mean, accuracy, precision, recall	Complete for balanced data set.
An iterative boosting-based ensemble algorithm [97]	It is based on boosting. It adds a suitable number of base learners each time a batch is acquired. The number of learners to be added depends on the accuracy of the ensemble on the last evaluated data batch.	Accuracy, computation cost	A possible limitation of the IBS algorithm may be related to the overgrowth of the number of base learners for some parameter settings, in specific domains.
A clustering and ensemble-based classifier algorithm [98]	It is an ensemble clustering method that contains ensemble boosting and clustering. Ensemble boosting is used for handling a large amount of data. Grid and density-based clustering are used as a base learner.	Time, accuracy, and memory	This method requires more time and memory as compared with the other algorithms.

2.7 Evaluation parameters for data stream classification

In the literature, various evaluation measures like accuracy, recall, precision, F1-score, G-mean, AUC were considered for the evaluation of data stream classifiers during the classification of data streams. However, accuracy is not always a perfect evaluation metric for performance evaluation, especially in an unbalanced class. So, there is a need to consider other evaluation measures for data stream classification. Table 2.9 shows the evaluation parameters used in different research works in data stream classification.

Table 2.9 Evaluation parameters

Evaluation parameter	Referred in
Accuracy	[23], [26], [27], [28], [31], [32], [33], [34], [35], [36]
Precision	[28], [30], [34]
Recall	[28], [30], [34]
F1-score	[28], [30], [34]
Computation time	[21], [26], [31], [32], [36]

2.8 Conclusion

This chapter summarized the literature relating to traditional ML techniques to classify data streams. The challenges and issues faced by ML algorithms in handling streaming data were identified. Further, the DL approach to improve the efficiency of data stream classifiers was reviewed. In conclusion, this review demonstrated the shortcomings of the traditional ML-based approach to classifying data streams. These studies also showed that the DL approach was effective and fast compared to the traditional ML-based approach.

Furthermore, the evidence reviewed here suggested that feature selection and the ensemble approach could further enhance the efficiency of the DL model. At the end of this chapter, evaluation parameters used by various research works in the area of data stream classification were also discussed. It can be concluded that evaluation parameters like accuracy are not enough to evaluate data stream classifiers, especially when the data set is unbalanced. Thus, data stream classifiers need to be evaluated using parameters applicable for balanced and unbalanced data streams.

Chapter 3

The Deep Ensemble Algorithm Learning (DEAL) Framework

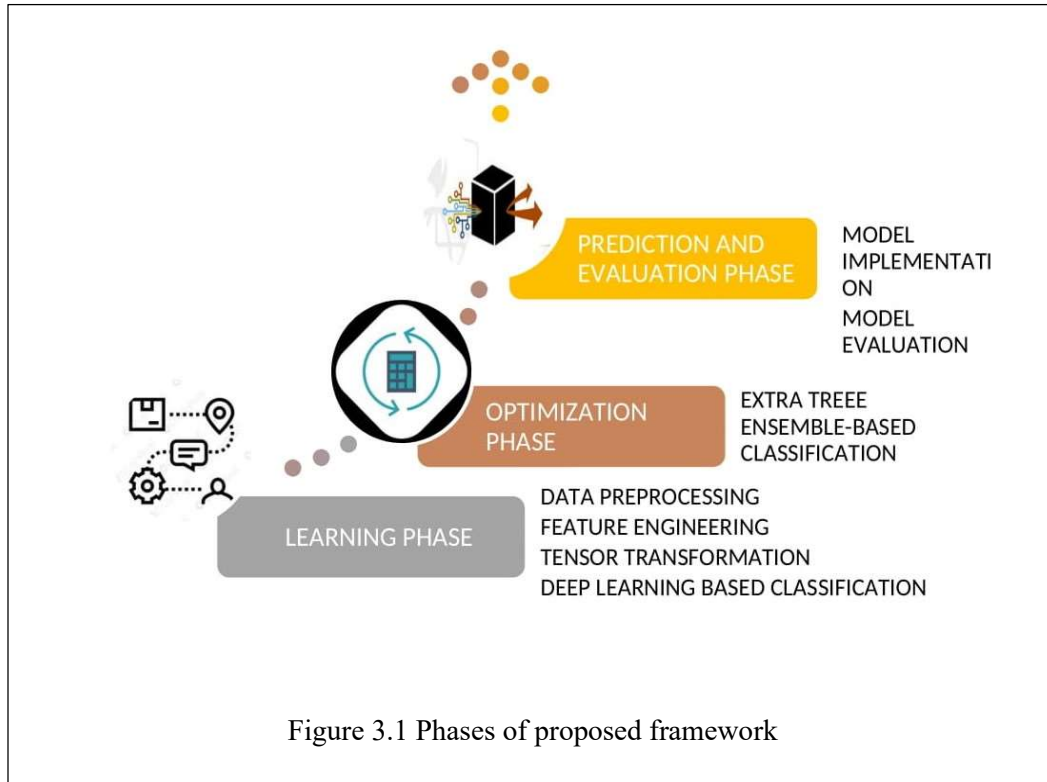
Data stream classification is crucial to take intelligent decisions on the fly. In the last decade, several studies and experiments have been conducted to explore efficient techniques for data stream classification. A study of relevant literature in Chapter 2 reveals that conventional ML classification techniques provides expected performance with static data. However, these conventional techniques encounter many challenges like handling concept drifts, multidimensionality, and a high volume of data in dynamic data stream environments. Furthermore, the performance of conventional ML classifiers degraded due to the transformation of data sets from static to dynamic data streams. Thus, researchers shifted their attention from conventional ML algorithms to DL based algorithms to better adapt to the dynamic environment. The survey of past work also exposed possibilities to improve the performance of the classifiers using techniques like ensemble approach, feature selection, and optimization. Therefore, the present research aims to design and implement an efficient data stream classifier based on deep learning and an ensemble-based approach to contribute in the same direction. The contribution of this research is manifold. The proposed framework predicts and classifies instances from real-time data streams. It can handle binary and multiclass classification and also handle classes of unbalanced data proportions.

The proposed Deep Ensemble Algorithm Learning (DEAL) framework is based on neuro-computational models and ensemble optimization. The DEAL combines deep learning for classification and extra tree (ET) ensemble for optimization to

perform predictions in real-time data streams. In Section 3.1, the construction and methodology of the proposed framework are presented. The section includes the phases/steps involved in defining the construction of the DEAL framework and the flowchart representing the overall methodology. Section 3.2 concludes the chapter and gives an overview of the next chapter.

3.1 A novel DEAL framework for efficient classification of data streams

This section illustrates the construction of the proposed framework for making predictions in real-time data streams[99]. The proposed framework works in three phases. The first phase is the learning phase. In this phase, the data is made ready for feeding in the DNN, and the DNN is trained with training data to classify data streams. The second phase is the optimization phase. In this phase, the output of the first phase is optimized with an extra tree ensemble-based optimization technique. Finally, the model is implemented to predict the class of the input data streams using testing data. In this phase, the implemented model is evaluated using metrics that suit the data stream environment and are often ignored in previous works. The proposed framework is adaptive for the changes in data streams and robust against incoming latent transaction patterns. Figure 3.1 shows the complete framework, and the following section explains the various components of the framework.



3.1.1 Learning phase

Learning is an offline phase. It involves data preparation, feature engineering, tensor transformation, and DL-based classification. Figure 3.2 shows the steps involved in learning phase.

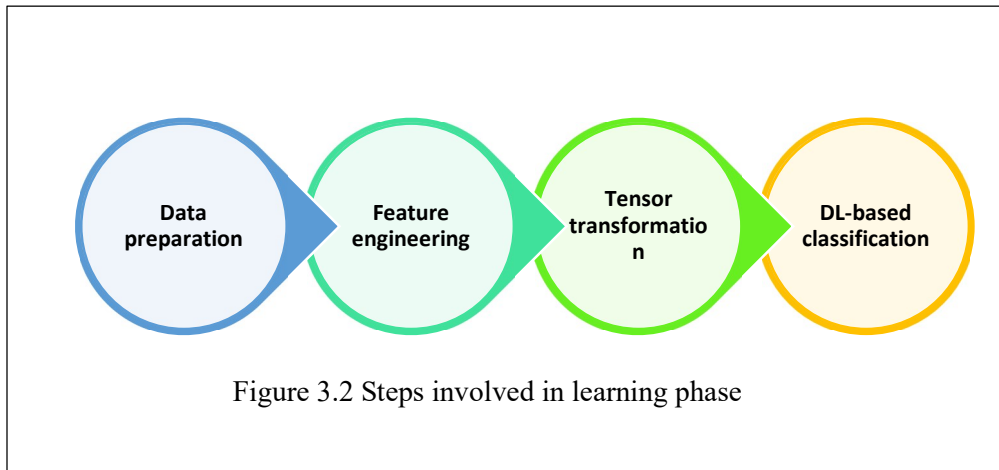


Figure 3.2 Steps involved in learning phase

The learning phase of the proposed model comprises the following steps:

1. **Data preparation and visualization:** Data preparation is preparing the data for feeding into an analytics platform. The data to be analyzed must be cleaned, structured, and transformed into a form that analytics tools can consume. The data preparation and visualization step of the proposed framework is shown in Figure 3.3.

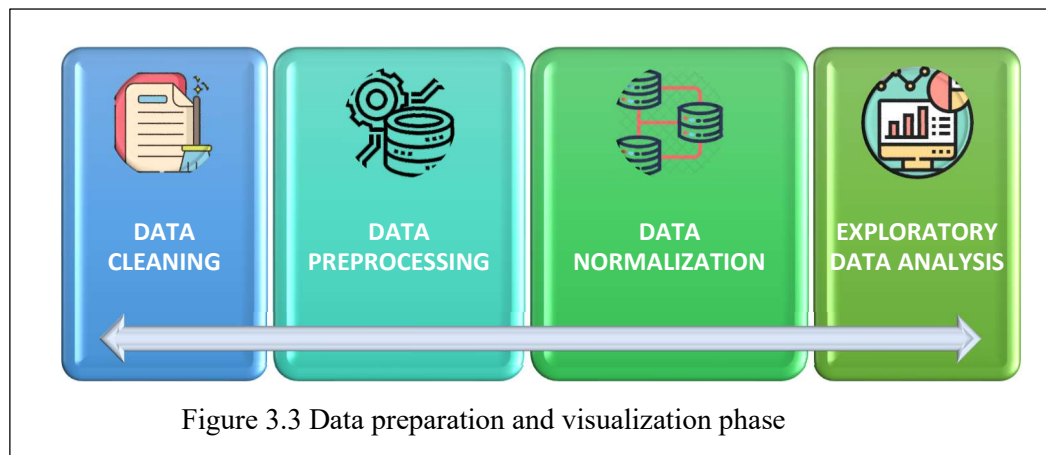


Figure 3.3 Data preparation and visualization phase

To prepare the data for analysis, data cleaning, preprocessing, normalization, and exploratory analysis are performed.

In this framework, initially, the data cleaning process takes place by consolidating/separating fields and columns, altering formats (nominal to categorical), eliminating obsolete or garbage data, and correcting the data. Once the data is cleaned, it enters into a preprocessing stage; at this stage, missing values, attributes, and outliers will get treated, and then the data is normalized to transform the data values to a standard scale. Normalization takes place without any distortion or loss of information. After data normalization, exploratory data analysis takes place. Exploratory data analysis is performed to discover patterns, detect anomalies, test hypotheses, and verify assumptions with the help of summary statistics and graphical representations.

2. **Feature engineering:** Feature engineering refers to the knowledge applied to create new features, select relevant features, and reduce data dimensionality. Feature engineering makes the data set compatible and improves the analytical model's performance. In this framework, the feature engineering step is carried out using different techniques depending on the data set and applications, like
 - For the datasets having missing values, Imputation (numerical and categorical imputation) is applied.
 - For categorical or numerical data, binning is applied to prevent model overfitting and to make the model robust.
 - For highly skewed data sets, Log transform is applied.
 - For changing categorical data to a numerical format One-hot encoding is done.

- 3. Tensor transformation:** A tensor is a potentially higher dimensional representation of vectors and matrices. Tensor algebra is a known technique for the training and operation of DL models. Tensor transformation enables the conversion of rich features in the model's format for easier and faster computation. In this framework, the features have to be transformed into tensors to fit into the model. The tensors are represented as n -dimensional arrays of base data types. Tensors are used to extract maximum performance from the system's hardware offering. The activations, pooling, convolution, and inner products evolve as primitives for optimized DL. Finally, correlation tensors are supplied as input to the DEAL model.
- 4. DL-based classification:** The DL model consists of input, output, and hidden layers. The first and last layers are input and output layers, respectively. This framework comprises a stack of "dense" layers, "activation" functions, and "dropouts." "Dense layer-1" accepts input features, nonlinear transformation operations take place at hidden layers, and the output layer presents the classification results. The architectural diagram shows a stack of "dense" layers, "activation" functions, and "dropouts." The architectural diagram for the proposed DL network is generated using the "matplotlib" Python library and is shown in Figure 3.4.

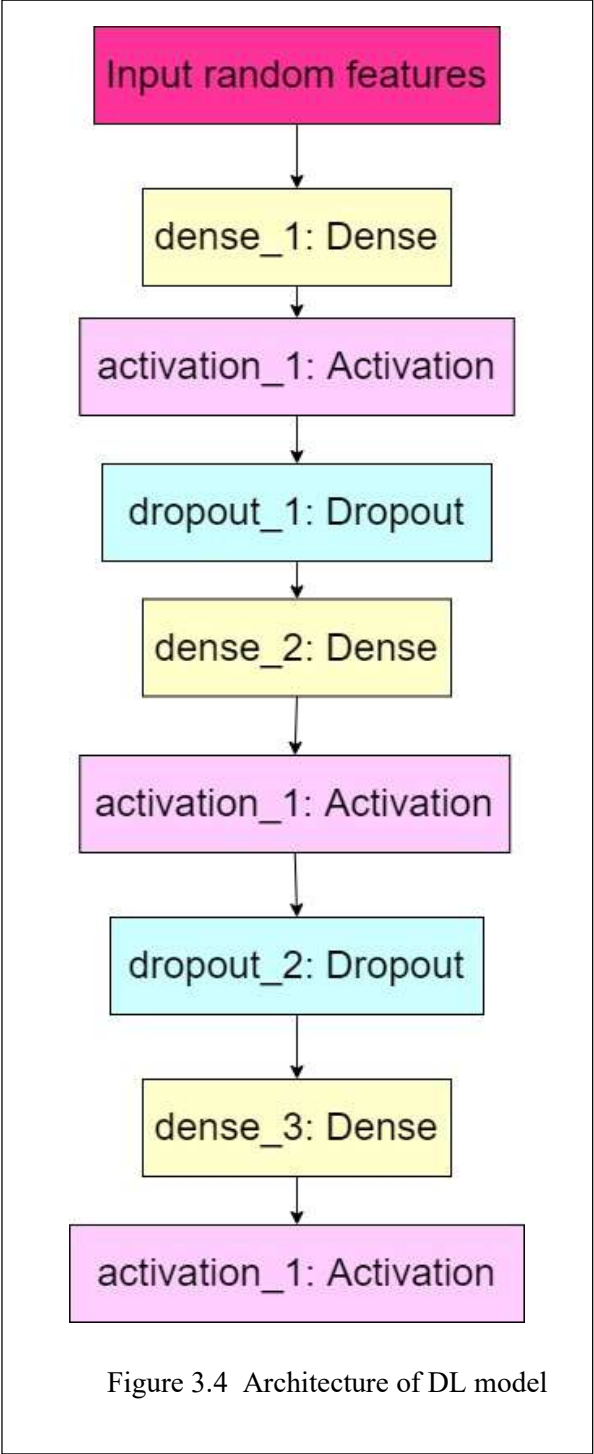


Figure 3.4 Architecture of DL model

Each new layer in the proposed model is a collection of nonlinear functions of the weighted sum of the previous layer's (fully connected) outputs, as shown in Equation 1.

$$f(x) = (\sum_{i=1}^m w_i * x_i) + b \dots \dots \dots (1)$$

Where,

m is the number of neurons in the previous layer

w is a random weight

x is the input value

b is the random bias

Further, in the proposed DL model, supervised learning takes place by adjusting the connection weights. The weights are adjusted for each neuron depending on the errors, outputs received, and expected results. The weight change Δw calculation uses Equation 2:

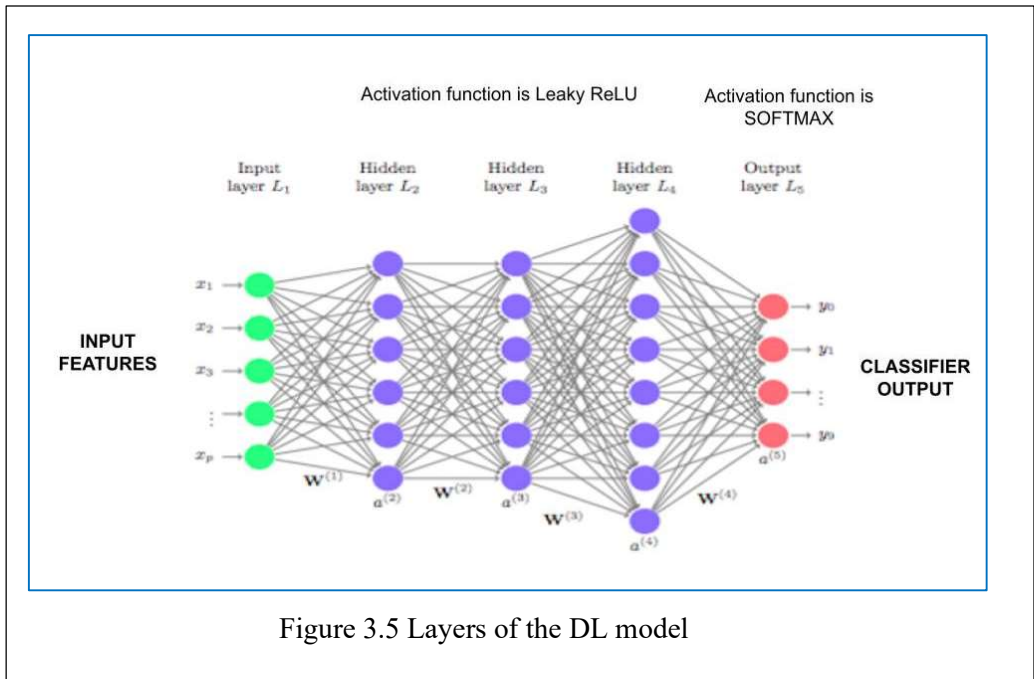
$$\Delta w = -\eta \partial E(W) / \partial W \dots \dots \dots (2)$$

Where η is the learning rate; $E(W)$ is the error function

The errors are sent back through back-propagation.

For binary classification, rectified linear unit (ReLU) activation function is used in dense layers 1 and 2, and sigmoid activation function is used in dense layer 3. Whereas, for multiclass classification, leaky ReLU activation function is used in

dense layers 1 and 2, and Softmax activation functions activation function is used in dense layer 3. In Figure 3.5, different layers of the DL model are shown.



The leaky ReLU has the advantage of overcoming the “dying ReLU” problem and thus takes less time to train a model. The mathematical expression for leaky ReLU is given in Equation 3.

$$f(x) = 0.01x \text{ for } x < 0 \text{ and } f(x) = x \text{ for } x \geq 0 \dots \dots \dots (3)$$

The details of the nodes, layers, activation functions, and dropouts are given in Table 3.1.

Table 3.1 Details of the nodes, layers, activation functions, and dropouts

Framework architecture	Hyper parameters: (CNN)	Hyper parameters: (MLP)	Hyper parameters: DEAL (proposed)
Input	Matrix	Matrix	Tensor
Convolutional layer	Filters: 8; ReLU	N/A	N/A
Max pooling (sub sampling) layer	Pool size: 2 or (2, 2)	N/A	N/A
Convolutional layer	Filters: 4; ReLU;	N/A	N/A
Connection layers	3	3	3
Dense layer	Nodes 32; ReLU	Nodes = 32: tanh	Nodes 32; ReLU
Dense layer	Nodes 16; ReLU	Nodes = 16: tanh	Nodes 16; ReLU
Output layer	2 Classes; Softmax	2 Classes; Score	2 Classes; Sigmoid, dropout = 0.1
Objective/Loss function	MAE	MAE	Categorical loss and MAE
Optimizer	SGD	SGD	ADAM
Max epochs	10	10 iterations	10
Validation split	0.2	N/A	0.1

Probability score is obtained as the output of the learning phase. The probability score is optimized during the optimization phase, to improve the efficiency of the proposed model.

3.1.2 Optimization phase

At this step, optimization of the DL model developed during the learning phase, takes place. The optimization improves the DL model’s performance and gets accurate predictions. In the proposed framework, the optimization is carried out using the extra tree ensemble feature selection technique. The ET ensemble algorithm is a top-down approach to build an ensemble of unpruned decision trees. The entire training sample is applied to split the nodes by selecting cut points randomly. The ET with k decision trees is used for feature selection. The value of k decides the number of features in a random sample of features. Here, the decision criteria used is information gain. Each feature is ordered in descending order according to the Gini importance of each feature, and the top k features are selected. In this way, a highly predictive feature subset is obtained.

The formula for information gain is given in Equation 4:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \dots\dots\dots(4)$$

The ensemble consists of several classifiers, each with its feature subset. The final feature subset is obtained by ensembling all the feature subsets. The ensemble is updated with a new classifier having an updated feature subset each time the performance drift occurs. The final prediction or classification is obtained by the majority vote among the aggregated predictions. In this way, a highly predictive feature subset is obtained. The feature selection using the extra tree ensemble approach is depicted in Figure 3.5.

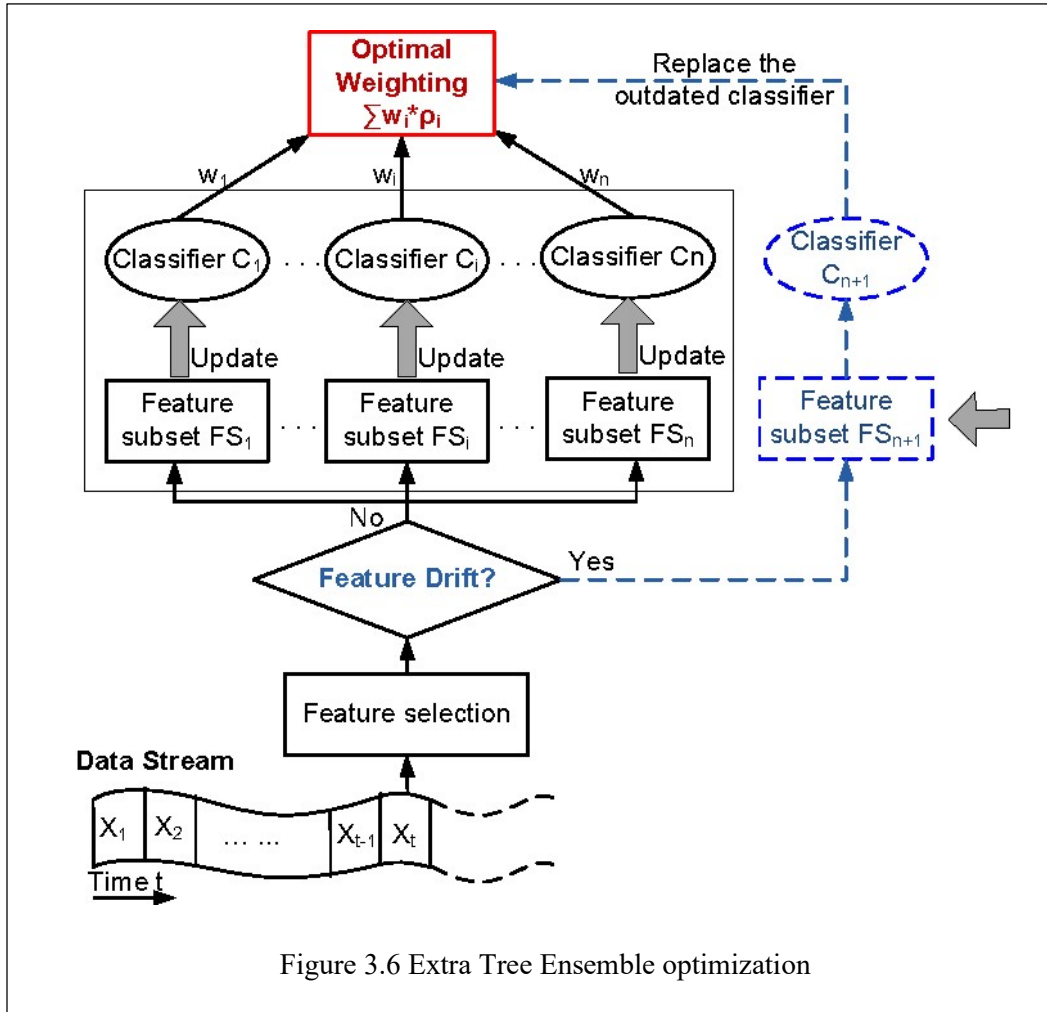


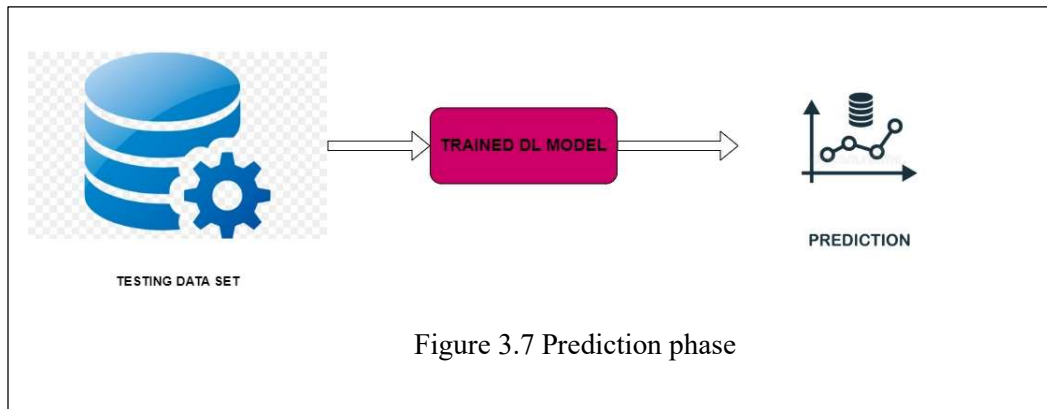
Figure 3.6 Extra Tree Ensemble optimization

Ensemble learning thus enhances the feature selection and performance of the classifier by selecting the most relevant features and removing irrelevant and redundant features. In addition, selecting the minimum number of features reduces the computational complexity of the model as only a small subset of the features is processed in deep learning. The DL model is thus optimized with ensemble-based features to maximize performance.

This approach achieves the approximately optimal global FS by acquiring feature subsets from different perspectives. Finally, the objective function, such as the cost or loss function of the model, is minimized to optimize the model.

3.1.3 Prediction phase

The developed and trained model in learning and optimization phases is deployed to classify various data streams in the prediction phase. The prediction phase is depicted in Figure 3.6.



In this work, the developed model is implemented in different domains like CCFD, stock trend predictions, HAR, and poker games. Different data streams are preprocessed using different techniques, and preprocessed data is fed into the trained model for prediction.

The deployed model is evaluated using various standard metrics like prediction accuracy, precision, recall, and F-score. In addition, the model is also evaluated using categorical accuracy. This evaluation metric is significant when the data stream is unbalanced. Figure 3.7 shows the flowchart for constructing the proposed framework.

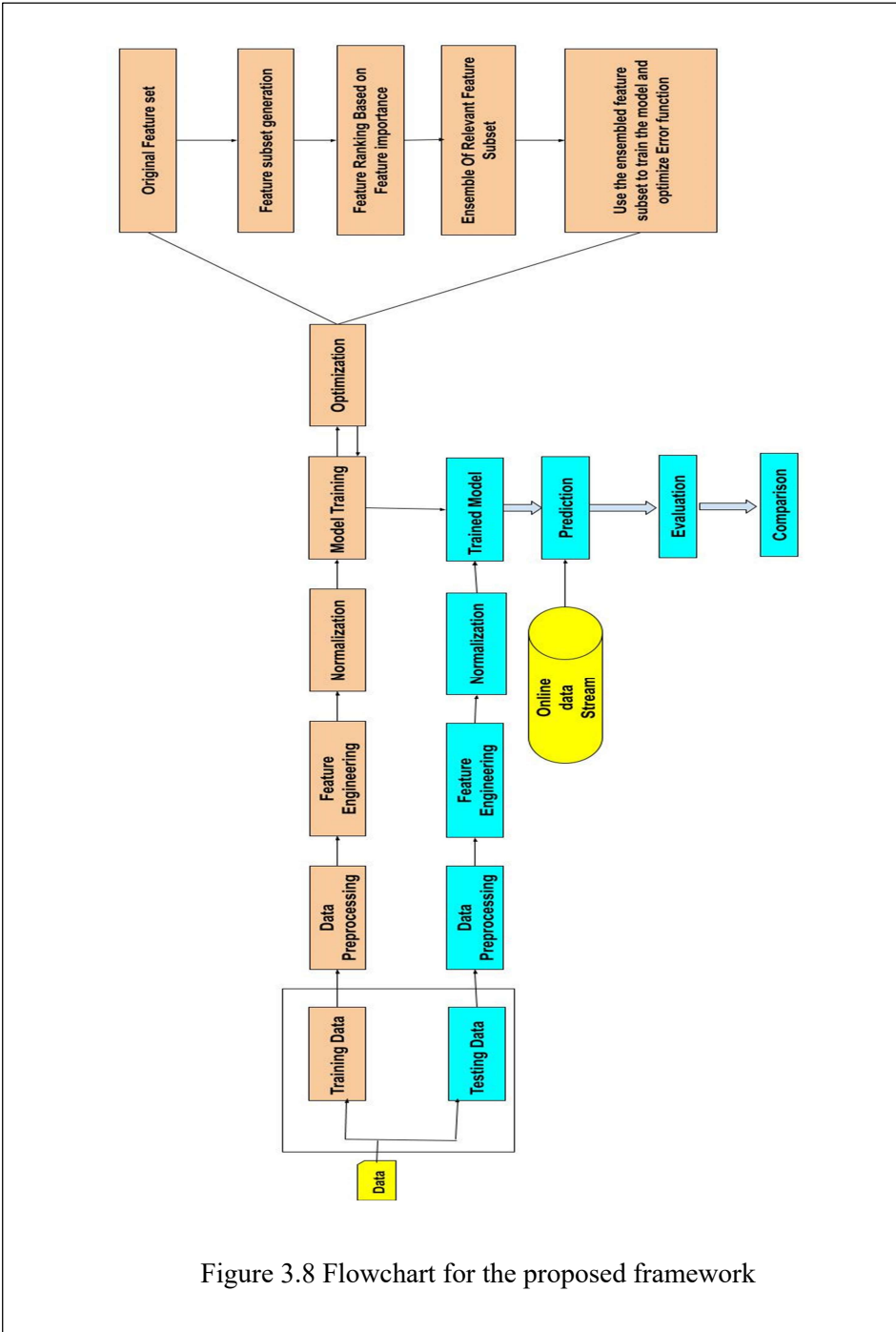


Figure 3.8 Flowchart for the proposed framework

3.2 Conclusion

In this chapter, the DEAL framework for real-time data stream classification has been presented. Further, the chapter also detailed all the three phases that are involved in developing the DEAL framework. The first phase is for learning and training the DEAL model. In this phase, the input data is initially cleaned, transformed, and finally fed into the DL model for classification. In the second phase, the model is optimized. ET ensemble feature selection technique is employed to optimize the proposed model for better prediction. Finally, the third phase is for prediction. In this phase, the trained model is used for classification of real-time data streams. In Chapter 4, details the algorithms and experimental setup for implementation of the DEAL framework has been presented.

Chapter 4

DEAL and Extra Tree Ensemble Algorithm for Data Stream Classification

The continuous generation of data streams is common in data-intensive applications like wireless sensor networks and social media platforms. Classification and analysis can glean meaningful information from these streams. However, it is difficult to extract significant knowledge from these potentially infinite data sets due to their changing nature and quick arrival rate. To overcome these issues, a DEAL framework for efficient data stream classification is proposed in detail in Chapter 3. This chapter presents the algorithms designed to implement the proposed DEAL framework. Various advanced ML libraries and tools have been used for experiments in this research work. The experimental setup, tools, and libraries used to implement the DEAL framework for real-time data streams are discussed in this chapter. There are many evaluation metrics like accuracy, false-positive rate, sensitivity, and so on to measure the performance of a classifier.

Furthermore, there are specific metrics for different sectors, as each region has different priorities and goals. This chapter also presents the evaluation parameters used in this research work. Section 4.1 presents the algorithms for the proposed DEAL framework. The experimental setup, including tools and libraries, is detailed in Section 4.2. The parameters for performance evaluation are discussed in Section 4.3. Finally, Section 4.4 concludes the chapter.

4.1 Algorithms for the proposed framework

The algorithms designed based on the proposed framework for efficient data stream classification are presented in this section. Two algorithms, DEAL and ET ensemble, are designed to implement the proposed framework. The DEAL algorithm implements the learning phase of the proposed approach, while the DL model is trained for classifying data streams. The ET ensemble algorithm implements the optimization phase, while the DL model is optimized for better predictions. The algorithms are discussed in detail in the following subsections 4.1.1 and 4.1.2.

4.1.1 DEAL Algorithm

The proposed framework uses the DEAL algorithm to implement the learning phase for training the DL model. The DL model is trained to classify the data streams in this phase. In the first step of the DEAL algorithm, features are supplied as input to the input layer of the DL model. These input features are transformed into tensors. The tensors are supplied as input to the dense layer of the DL model. In the dense layers (one, two, and three), the nonlinear function of the weighted sum gets calculated, and activation function ReLU is applied to the output. During this process, the output of the current layer becomes the input of the next higher layer. Finally, the sigmoid activation function is applied in the output layer for binary classification, and Softmax is applied for multiclass classification. The extended form of ReLU—that is, leaky ReLU—has been used in this work to speed up the training process. In the output layer, the probability score is calculated. This score gives the probability of an instance belonging to a particular class. The output of this phase is optimized in the next phase. Table 4.1 lists the notations used in the DEAL algorithm, followed by the DEAL algorithm in Algorithm 1.

Table 4.1 Notations for DEAL algorithm

Notations:

T_{x_n} , T_{d_n} , and T_{d_m} are tensors applied to input, dense layer 1, and dense layer 2, respectively.

W_n , W_m , and W_o are activations applied at dense layer 1, dense layer 2, and output, respectively.

$$x_i = \{x_1, x_2 \dots \dots x_n\}$$

$$w_i = \{w_1, w_2 \dots \dots w_n\}$$

x_i is feature; w_i is weight; n is no. of features/weights.

b_n , b_m , and b_f , are biases applied at dense layer 1, dense layer 2, and output layer

ReLU:

$$relu(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

σ : Sigmoid function:

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

\hat{y} : Score calculated by DL

γ : Score calculated by an ensemble

$h(\cdot)$: Ensemble

Algorithm 1: DEAL algorithm

Start

1. Input Features: \mathbf{x}_n

2. Transform it to tensor: \mathbf{T}_{x_n}

3. At dense layer 1, feed \mathbf{T}_{x_n}

4. Calculate: $[\mathbf{T}_{x_n}, \mathbf{w}_n] = \mathbf{T}_{x_n} * \mathbf{w}_n$

5. Apply *relu* activation function to output tensor $[\mathbf{T}_{x_n}, \mathbf{w}_n]$:

$$\mathbf{T}_{d_n} = \mathit{relu}(\mathbf{W}_n[\mathbf{T}_{x_n}, \mathbf{w}_n] + \mathbf{b}_n)$$

6. At dense layer 2, feed: \mathbf{T}_{d_n}

7. Calculate: $[\mathbf{T}_{d_n}, \mathbf{w}_m] = \mathbf{T}_{d_n} * \mathbf{w}_m$

8. Apply *relu* activation function to output tensor $[\mathbf{T}_{d_n}, \mathbf{w}_m]$

$$\mathbf{T}_{d_m} = \mathit{relu}(\mathbf{W}_m[\mathbf{T}_{d_n}, \mathbf{w}_m] + \mathbf{b}_m)$$

9. At dense layer 3 (output), feed \mathbf{T}_{d_m}

10. Apply sigmoid activation function σ for binary or Softmax activation function for multiclass classification to \mathbf{T}_{d_m}

11. Calculate the probability score for predicting the class of transaction:

$$\hat{y} = \sigma(\mathbf{W}_o[\mathbf{T}_{d_m}] + \mathbf{b}_f)$$

12. Apply ensemble $\boldsymbol{\gamma} = \mathbf{h}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$

13. Finally, using an objective function such as error function: $E(\mathbf{W})$

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \log(\gamma) + (1 - y_i) \log(1 - \gamma)$$

14. Apply optimization

Output: Class labels

4.1.2 ET Ensemble Algorithm

The optimization phase of the proposed framework uses the ET ensemble algorithm. The optimization minimizes the cost/loss function and improves the performance of the DL model. The DL model is optimized using the ET feature ensemble technique in this work. The idea behind this technique is that the performance of the data stream classifiers degrades due to changing concepts with time. The ET feature ensemble technique generates a new feature subset containing only relevant features required for data stream classification. The model is trained using the updated feature subset, and thus, the overall efficiency of the model improves. The first step checks for feature change in the current feature set in the ET ensemble algorithm. The new feature subset will be generated only if the current feature set has changed due to concept drift. For generating the new feature subset, the feature importance of each feature is calculated, and then the features are arranged in descending order of feature importance value. In the next step, the first k features are selected to form the feature subset, where k comprises the randomly selected features at each node. Finally, the feature subsets are ensembled to obtain the final feature subset. The ensembled feature subset incorporates the new concepts. Table 4.2 lists the notations used in algorithm 2 followed by ET algorithm in Algorithm 2.

Table 4.2 Notations for ET ensemble algorithm

Notations:

$sAN(S)$: split at node S method

S : the training subset corresponding to the output class (node).

f : is feature split.

f_c : is candidate features or

f_k : random cut points returned by chooseRS(S, f) method

$stopS(S)$: is stop split method with parameter S

K : no. of features randomly selected at each node.

$chooseRS(S, f)$: Choose a random split method

fS_{min} : Minimal value of f in S

fS_{max} : Maximal value of f in S

$stopSplit(S)$: stop split method

n_{min} : is the minimum sample size for a split.

Algorithm 2: ET ensemble algorithm

sAN(S)

Input: S

Output: a split $[f < f_c]$ or none

1. if (stopS(S)) is TRUE

 { return None }

else

{

 a) select K features $\{f_1, \dots, f_k\}$ from all f_c that non constant (in S);

 b) Take K splits $\{s_1, \dots, s_K\}$, where $s_i = \text{chooseRS}(S, f_i) \forall_i = 1, \dots, K$;

 c) return a split s^* such that $\text{Score}(s^*, S) = \max_{I=1, \dots, K} \text{Score}(s_i, S)$.

}

chooseRS (S, f)

Inputs: S and f

Output: a split

1. Draw a random cut-point f_k uniformly in $[f_{S_{\min}}, f_{S_{\max}}]$;
2. return the split $[a < a_c]$.

stopSplit (S)

Input: S

Output: a boolean

1. if ($|S| < n_{\min}$) then
 return TRUE;
 2. if all features are constant in S,
 return TRUE;
 3. if the output is constant in S,
 return TRUE;
- else
- return FALSE.

The following section presents the system requirements, tools, and libraries used for the DEAL framework implementation.

4.2 System requirements, Tools, and Libraries

The experimental system comprises an Intel Core i5 (2.0 GHz.) processor with 8 GB RAM. The operating system environment is Microsoft Windows 10 (64 bit). The system is installed with DL libraries: Keras, Google TensorFlow, and other useful libraries such as Scikit-Learn, NumPy, and Pandas installed over Python. The Python language is a well-known object-oriented, interactive, and dynamically typed programming language. The model is developed in Python language (version

3.7.0 and 3.6.6) utilizing Keras as the high-level API developed for Google's TensorFlow. The system details are mentioned in Table 4.3.

Table 4.3 System details

System/Tools/Environments/Libraries	Version/Configuration
Processor	Intel Core i5 10th generation
RAM	8 GB
Operating System	Windows 10 (64 bit)
Python	3.6.6 and 3.7.0

Tools and Libraries

The tools and libraries used to carry out the implementation of the proposed framework are as follows:

Tools

Anaconda Navigator is a desktop GUI to launch applications and manage packages needed for experimentation.

Jupyter Notebook is open-source software to create and share live code, equations, visualizations, and narrative text within a document.

Google Colaboratory, or “Colab” for short, is a product of Google Research. Colab allows anyone to write and execute arbitrary Python code through a browser and is particularly suitable for ML, data analysis, and education.

Libraries

TensorFlow for high performance and flexible numerical computation with solid support for ML and DL across many scientific domains.

Keras is a high-level NN API, written in Python and capable of running on top of TensorFlow.

Pandas is a powerful data-frame (tabular) object for data structures and analysis.

Seaborn and **matplotlib** are used, which are popular graph libraries for comprehensive 2D plotting.

Scikit-learn is used, which provides simple and efficient tools for data mining and analysis. It is a powerful tool built on SciPy, NumPy, and matplotlib.

Scipy is used, which is a fundamental package for mathematics and scientific computing.

Numpy is a fundamental package for numeric computation.

Table 4.4 gives a detailed description of tools and libraries.

Table 4.4 Tools and libraries

Tools/libraries	Version/configuration	Brief description
Anaconda Navigator	v_5.3.1; x86_64 bit	Anaconda Navigator is a desktop GUI to launch applications and manage packages needed for experimentation.
Jupyter Notebook	v_5.7.2; 64 bit	Open-source software to create and share live code, equations, visualizations, and narrative text within a document.
Python	3.7.0 and 3.6.6	A well-known object-oriented and interactive, and dynamically typed programming language.
Scikit-Learn	0.20.2	Provides simple and efficient tools for data mining and analysis. It is a powerful tool built on SciPy, NumPy, and matplotlib.
SciPy	1.1.0	Fundamental package for mathematics and scientific computing.
NumPy	1.15.4	Fundamental package for numeric computation.
TensorFlow	1.12.0	TensorFlow for high performance and flexible numerical computation with strong support for machine learning and deep learning across many scientific domains.
Keras	2.2.4	Keras is a high-level NN API, written in Python and capable of running on top of TensorFlow.
Pandas	0.23.4	Powerful data-frame (tabular) object for data structures and analysis.
Seaborn and Matplotlib	0.9.0 and 3.0.2, respectively	Popular graph libraries for comprehensive 2D plotting.

4.3 Evaluation parameters

The confusion matrix for two-class and multiclass classification is the most intuitive and easiest way to find the performance of the model.

Confusion Matrix

		Predicted	
		TP	FN
Actual	TP	TP	FN
	FP	FP	TN

The metrics used to evaluate the performance of the proposed model are accuracy, precision, recall, F1-score, and categorical accuracy. In this work, categorical accuracy, which is often ignored in previous works, is used. These parameters have been obtained using the confusion matrix.

Categorical Accuracy

Accuracy might be deceiving when dealing with a skewed data set. A model can accurately forecast the value of the majority class in all predictions and attain a high classification accuracy in such a situation. However, the model is ineffective in the problem domain because of the significant class imbalance. The categorical accuracy is crucial in evaluating data stream models, especially with an unbalanced data stream.

The categorical accuracy metric determines how well the model makes the correct prediction. It is the percentage of predicted values that match up with actual values. Therefore, categorical accuracy = 1 indicates that the model's predictions

are entirely accurate. The details of the evaluation parameters are given in Table 4.5.

Table 4.5 Evaluation parameters

Evaluation parameter	Description	Formula
Accuracy	Classification accuracy is a metric that summarizes the performance of a classification model as the number of correct predictions divided by the total number of predictions.	$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$
F1-score	F1-score is a measure of a model's accuracy on a data set.	$F1\text{-score} = \frac{2 * Precision * Recall}{Precision + Recall}$
Recall	Recall is a measure of the classifier's completeness—the ability of a classifier to correctly find all positive instances.	$Recall = \frac{TP}{TP + FN}$
Precision	Precision is the ability of a classifier not to label an instance positive that is actually negative.	$Precision = \frac{TP}{TP + FP}$
Categorical accuracy	It calculates the mean accuracy rate across all predictions for categorical classification problems.	$Categorical\ accuracy = \frac{Correct\ predict}{Total\ instances} * 1$

The parameters discussed here are used to evaluate the performance of the proposed DEAL framework.

4.4 Conclusion

In this chapter, the algorithms for implementing the DEAL framework have been presented with the system requirements, tools, and libraries required to implement the DEAL framework. The chapter has also described the metrics for performance evaluation of the proposed model. The next chapter presents the experimental results of the DEAL framework for binary and multiclass classification data streams.

Chapter 5

DEAL Framework for Data Stream Classification

For efficient data stream classification, Chapter 4 has proposed a DEAL framework. Further, that chapter also detailed the experimental setup and evaluation parameters for the DEAL framework. The present chapter details the experiments carried out on the DEAL framework for data stream classification. Both the binary and multiclass datasets is used for experimentation purpose. The dataset description along with the results and, comparison of prediction and categorical accuracy is also detailed. Section 5.1 presents the implementation results of the DEAL framework for data stream classification. Finally, Section 5.2 concludes the chapter.

5.1 Implementation of DEAL framework for data stream classification

In this section, the proposed DEAL framework is implemented for data stream classification. Chapter 4 has detailed the experimental setup along with the hyper-parameters. Experiments were carried out on both binary and multi-class datasets. The binary datasets used for the experiments are CCFD, stock prediction, SEA generator and hyperplane. The multi-class datasets used for experiments are HAR, poker hand, RBF and LED generator. In the experiments, the datasets were divided in the ratio of 80:20 for training and testing purpose. The DEAL framework was trained using the training dataset during the learning phase and in the prediction phase, testing dataset has been used for making predictions. The following subsections present the results of the implementation for each data set along with the prediction accuracy and categorical accuracy. The two accuracies were

compared to evaluate the performance of the DEAL framework in dynamic and imbalanced data stream scenario.

5.1.1 DEAL framework implementation on credit card dataset

The credit card data set is highly unbalanced, having 492 fraudulent transactions out of 284,807 transactions in contrast to 284,315 legitimate transactions[40]. Fraudulent transactions in the data set represent only 0.172% of the total transactions.

Figure 5.1 shows the distribution of normal and fraudulent transactions.

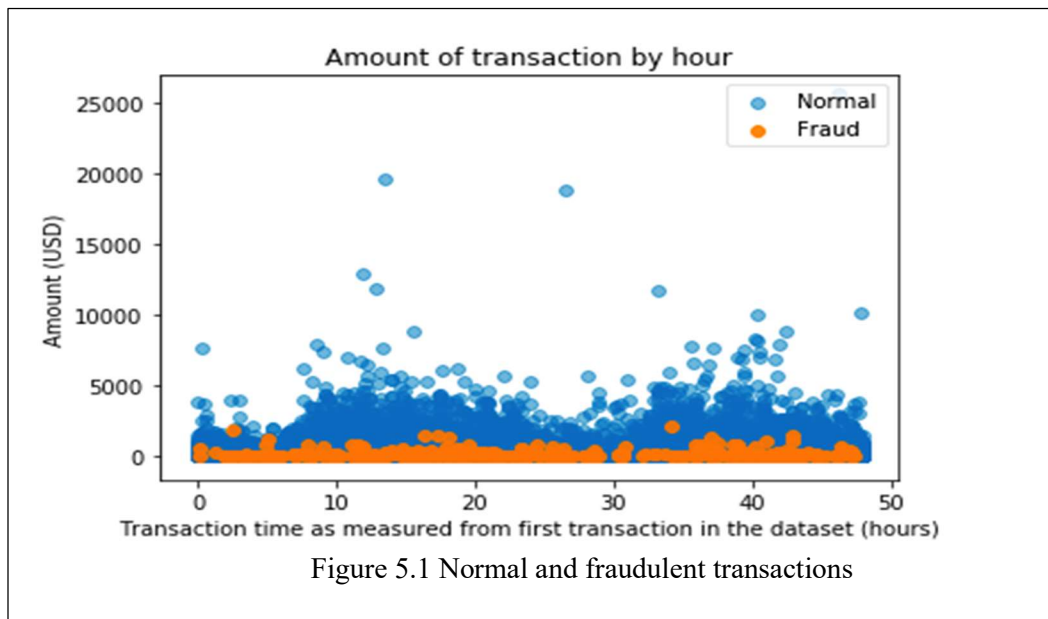
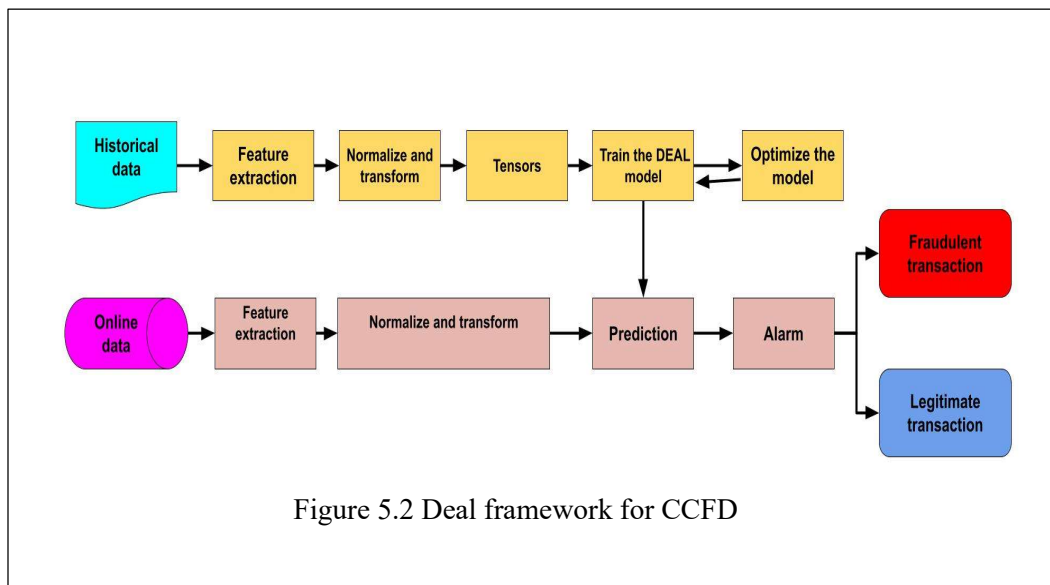


Figure 5.1 Normal and fraudulent transactions

This data set consists of 31 attributes, including the class attribute. Feature “class” is the target variable, and it is represented by a value of 1 in case of fraud and 0 in case of a standard transaction.

For experimentation purpose the dataset was divided into training and testing dataset, in the ratio of 80:20 i.e., 227846 instances to train the model and 56961 instances to test the model.

During the learning phase the DEAL framework trained the DL model over the training dataset. The DEAL algorithm thus iterated for all 227846 instances of training dataset. During the prediction phase, the trained DL model was used for prediction[100]. Figure 5.2 shows the implementation procedure of the DEAL framework for credit card fraud detection.



The output of the prediction phase determines whether the transaction was fraudulent or not. The results are summarized in Table 5.1.

Table 5.1 Results of DEAL over credit card fraud detection data set

S. No.	Evaluation parameter	Result
1	Prediction accuracy	99.81
2	F1-score	0.998
3	Recall	0.999
4	Precision	0.999
5	Categorical accuracy	99.79

An accuracy measure is not enough to evaluate the performance of the classifiers handling such a highly unbalanced data set. In this work, categorical accuracy measure is used for evaluation in unbalanced scenarios, which was ignored in previous works. The graph in Figure 5.3 compares the prediction accuracy and categorical accuracy of DEAL over the credit card data set.

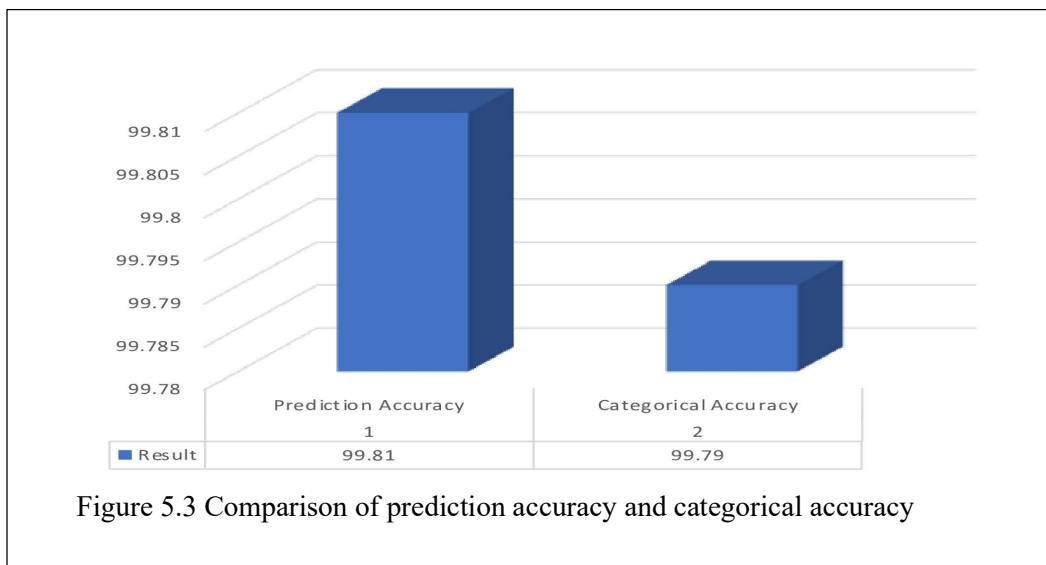
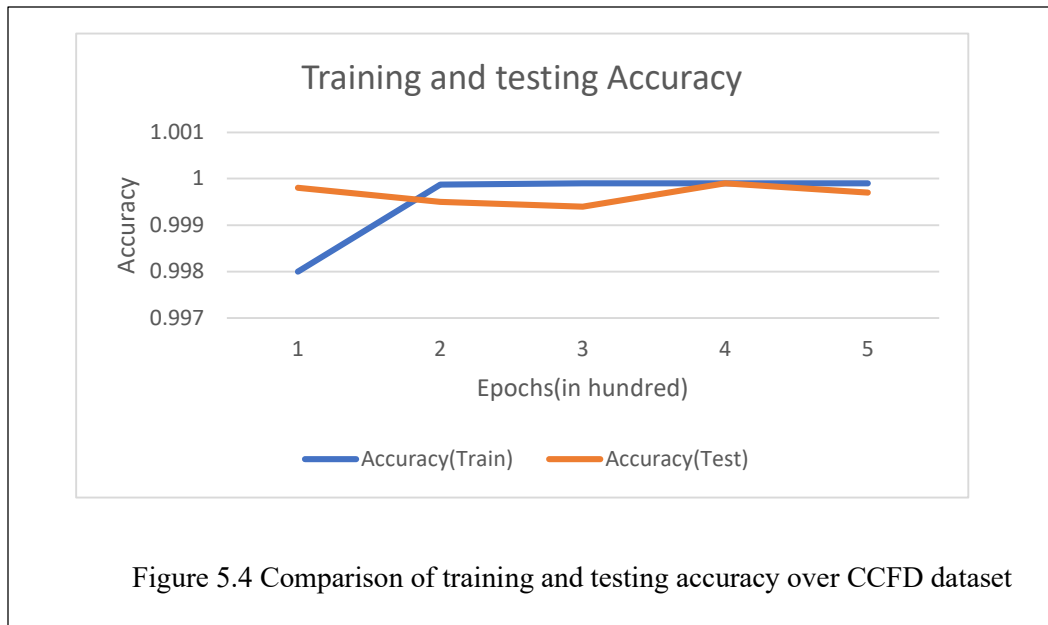


Figure 5.3 Comparison of prediction accuracy and categorical accuracy

The values of accuracy and categorical accuracy are high, indicating that the unbalanced data set is classified efficiently using the DEAL framework.

The graph in Figure 5.4 compares the training and testing accuracy of DEAL over the credit card data set.



The graph in figure 5.4 shows that the value of training and testing accuracies are approximately equal, thus there is no instance of overfitting or underfitting.

5.1.2 DEAL framework implementation on stock prediction data set

The stock indices data was acquired from the NSE website for a six-year duration (13th Apr. 2013 to 29th Mar. 2019) [101] [102]. For experiment purpose, bank, automobile, and metal indices were considered. The acquired dataset has 1463 instances 7 attributes. The training and testing data were numerical and was in the ratio of 80:20.

The proposed DEAL framework was implemented for predicting price trends of stock indices[103]. Most correlative and highly predictive stock technical indicators (STIs) were supplied as input to the DL model in the learning phase. During feature extraction, technical analysis (TA) was applied to derive the STIs. In the optimization phase, the output of the DL model was optimized. The optimization was done by minimizing the cross-entropy loss. During prediction, the target label was generated as a binary response attribute for binary classification. The values in target attributes indicated the buy (1) or sell (0) decision. Thus, the model predicts the daily signal for buying/selling the indices at the end of the day. Figure 5.5 shows the implementation of the DEAL framework for stock price prediction.

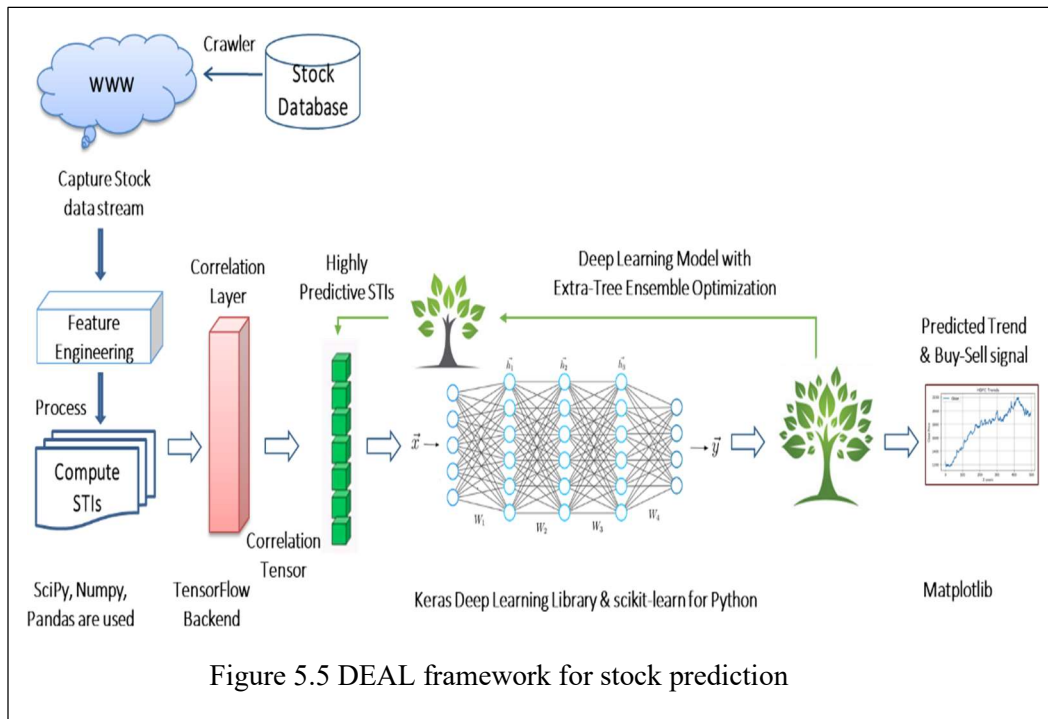


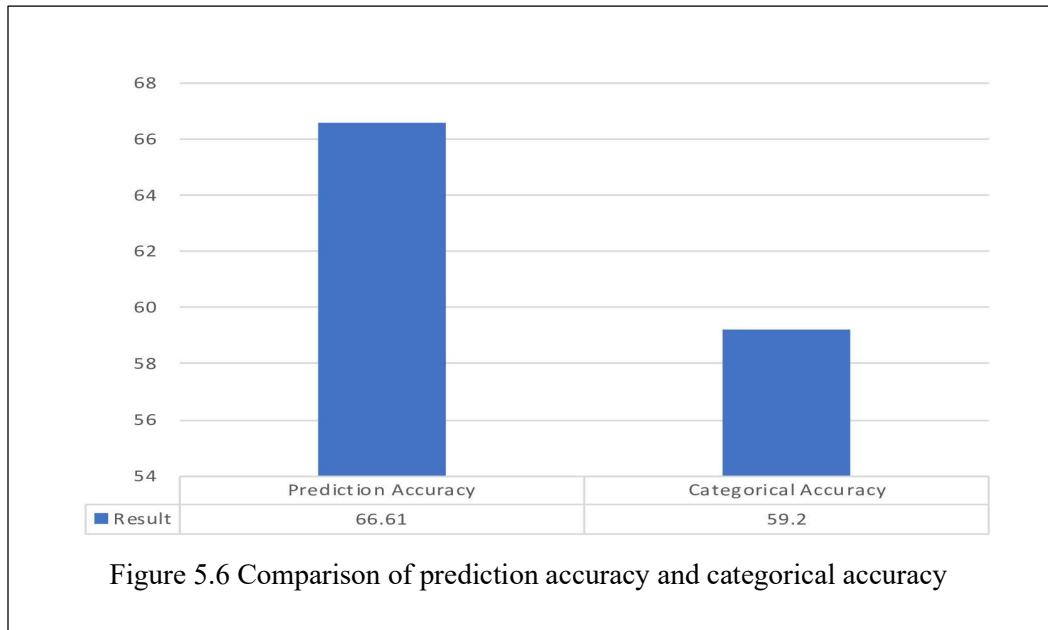
Figure 5.5 DEAL framework for stock prediction

The results of implementing the DEAL framework over the stock prediction data set are summarized in Table 5.2.

Table 5.2 Results of DEAL over stock prediction data set

S. No.	Evaluation parameter	Result
1	Prediction accuracy	66.61
2	F1-score	0.561
3	Recall	0.69
4	Precision	0.453
5	Categorical accuracy	59.2

The graph in Figure 5.6 compares the prediction accuracy and categorical accuracy of DEAL over the stock prediction data set.



5.1.3 DEAL framework implementation on hyperplane data set

The hyperplane generator is a synthetic dataset. It has 10 dimensions, two classes. For experimentation purpose, 10,0000 instances of dataset were generated. The training and testing dataset divided in the ration of 80:20. Table 5.3 shows the implementation results of the DEAL framework over the hyperplane data set.

Table 5.3 Results of DEAL over hyperplane data set

S. No.	Evaluation parameter	Result
1	Prediction accuracy	92.56
2	F1-score	0.91
3	Recall	0.79
4	Precision	0.87
5	Categorical accuracy	89.7

Figure 5.7 compares the prediction accuracy and categorical accuracy of DEAL over the hyperplane data set.

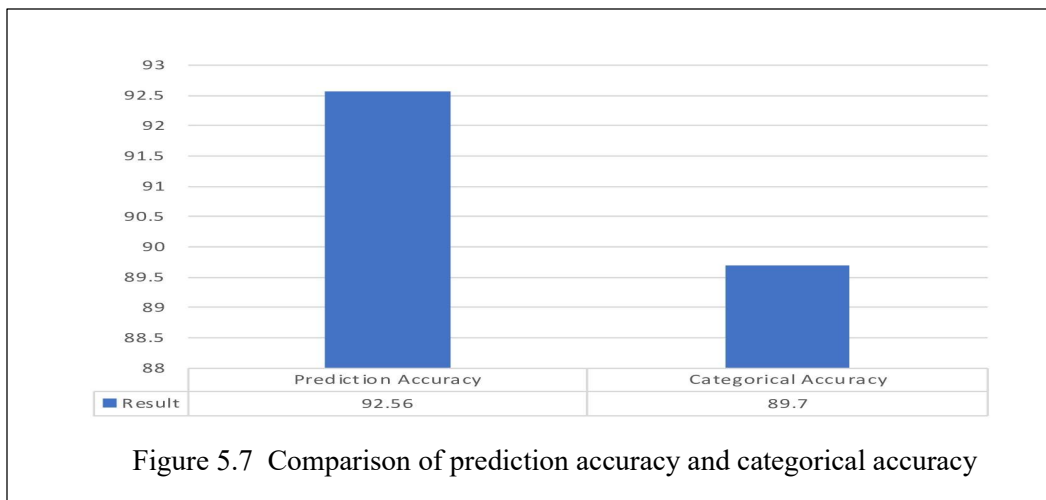


Figure 5.7 Comparison of prediction accuracy and categorical accuracy

5.1.4 DEAL framework implementation on sea generator data set

The SEA generator is a synthetic dataset. It consists of 50,000 instances with three attributes, of which only two are relevant. In experiment, the training and testing data divided in the ration of 80:20 i.e., 40,000 instances used to train the model and 10,000 instances used during prediction phase. Table 5.4 shows the implementation results of the DEAL framework over the SEA data set.

Table 5.4 Results of DEAL over SEA generator data set

S. No.	Evaluation parameter	Result
1	Prediction accuracy	93.76
2	F1-score	0.86
3	Recall	0.81
4	Precision	0.87
5	Categorical accuracy	90.03

The graph in Figure 5.8 compares the prediction accuracy and categorical accuracy of DEAL over the SEA generator data set.

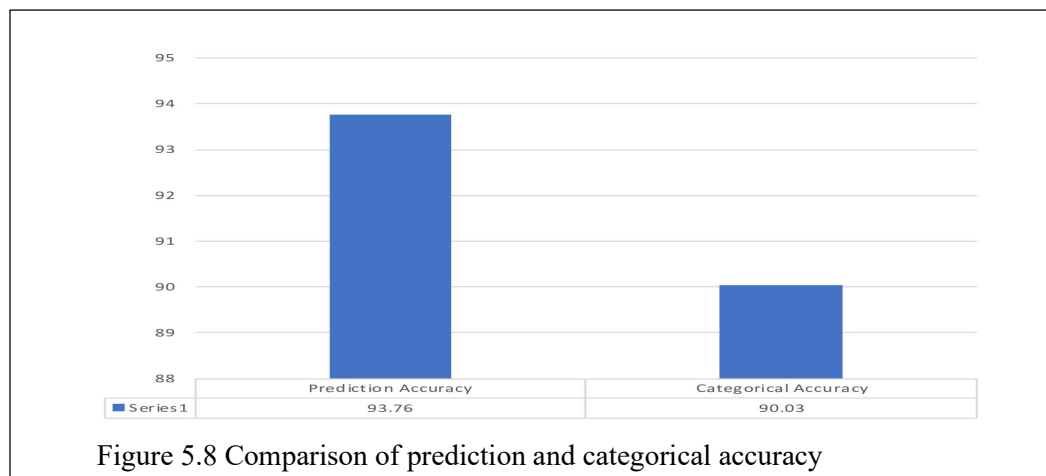


Figure 5.8 Comparison of prediction and categorical accuracy

5.1.5 DEAL framework implementation on HAR data set

The HAR data set contains recordings of 30 people aged 19–48 doing daily-life activities (standing, sitting, lying down, walking, and going upstairs and downstairs) while wearing a wearable device with inertial sensors. There are 7352 instances and six attributes.

The activity set is listed Table 5.5. Labels are used to represent the activity corresponding to the movements of the user.

Table 5.5 Activities in HAR data set

Label	Activity	Number of Instances
1	Lying down	1,407
2	Standing	1,374
3	Sitting	1,286
4	Walking	1,226
5	Walking upstairs	1,073
6	Walking downstairs	986

Figure 5.9 shows the percentage of different activities in the data set during observation.

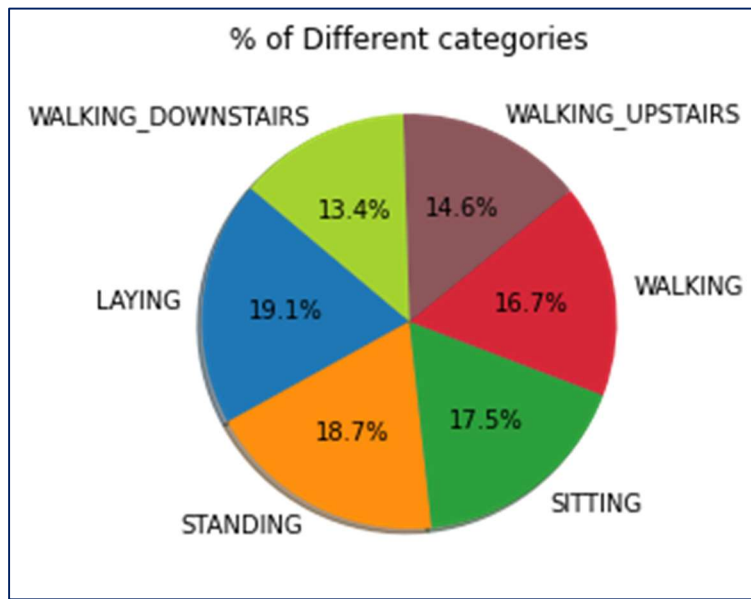


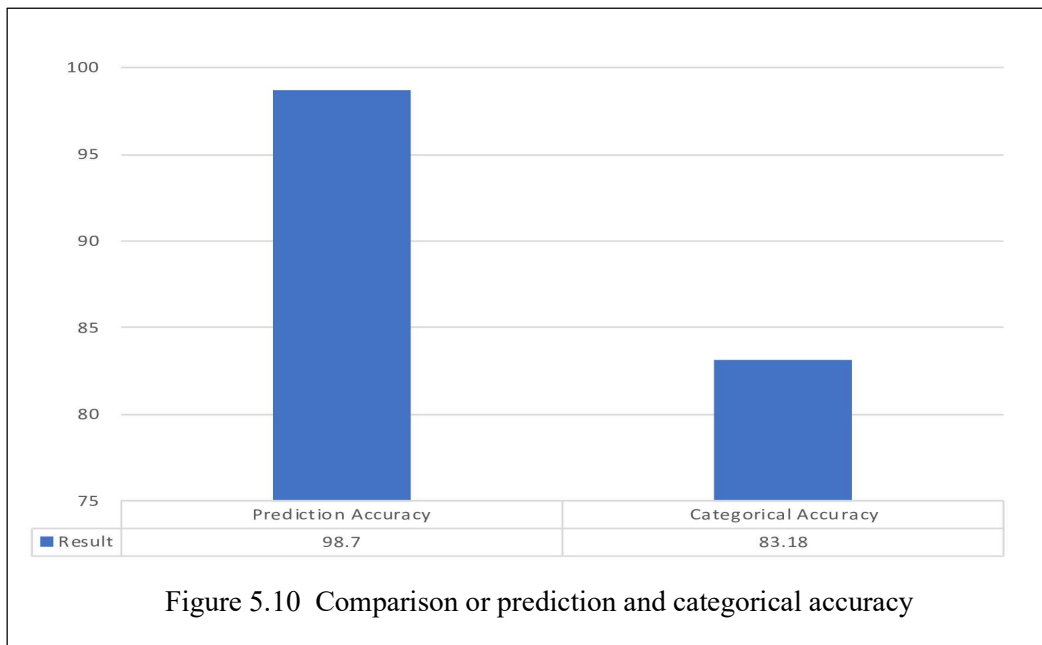
Figure 5.9 Percentage of different activities

The DEAL was implemented over the HAR data set to predict the different activities (multiclass classification). The training and testing data divided in the ratio 80:20 i.e., 5882 instances used for training the model and 1470 used for prediction. The DL model performs the classification and was trained on training data. The output of the DL model was optimized to improve the model's performance. The experiment results on HAR data set are summarized in Table 5.6.

Table 5.6 Results of DEAL over HAR data set

S. No.	Evaluation parameter	Result
1	Prediction accuracy	98.7
2	F1-score	0.989
3	Recall	0.995
4	Precision	0.981
5	Categorical accuracy	83.18

The graph in Figure 5.10 compares the prediction accuracy and categorical accuracy of DEAL over the HAR data set.



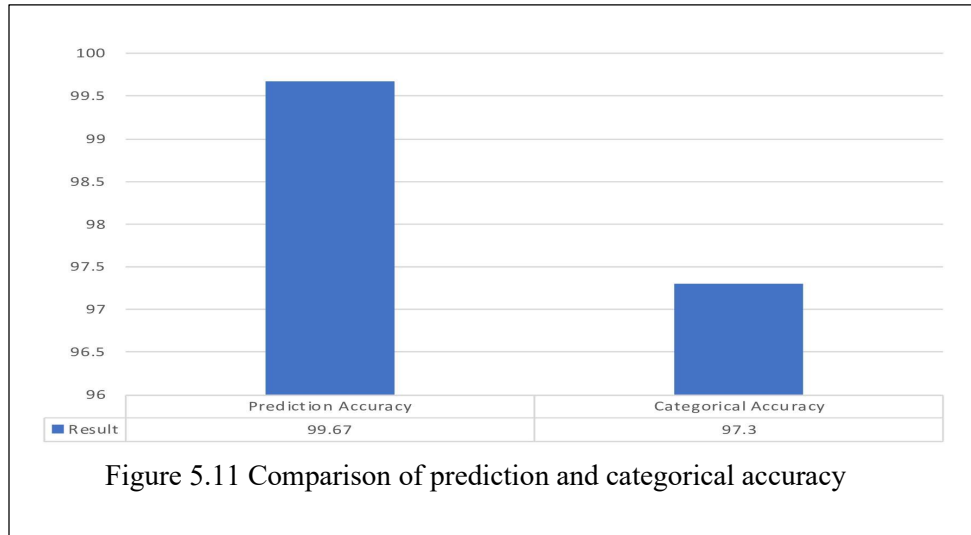
5.1.6 DEAL framework implementation on poker hand data set

The UCI repository poker data set has 11 attributes, 1,025,010 instances, and 10 classes [3]. Each record is an example of a hand consisting of five playing cards drawn from a standard deck of 52 cards. Each card is described using two attributes (suit and rank) for 10 predictive attributes. In addition, there is one class attribute that describes the “poker hand.” The dataset was divided in the ratio 80:20 for training and prediction phase i.e., 820008 instances used for training the DL model during learning phase and 205002 instances used in prediction phase. Table 5.7. shows the evaluation results of the research algorithm DEAL on the poker data set.

Table 5.7 Results of DEAL over poker hand data set

S. No.	Evaluation parameter	Result
1	Prediction accuracy	99.67
2	F1-score	0.74
3	Recall	0.65
4	Precision	0.82
5	Categorical accuracy	97.3

The graph in Figure 5.11 compares the prediction accuracy and categorical accuracy of DEAL over the poker hand data set.



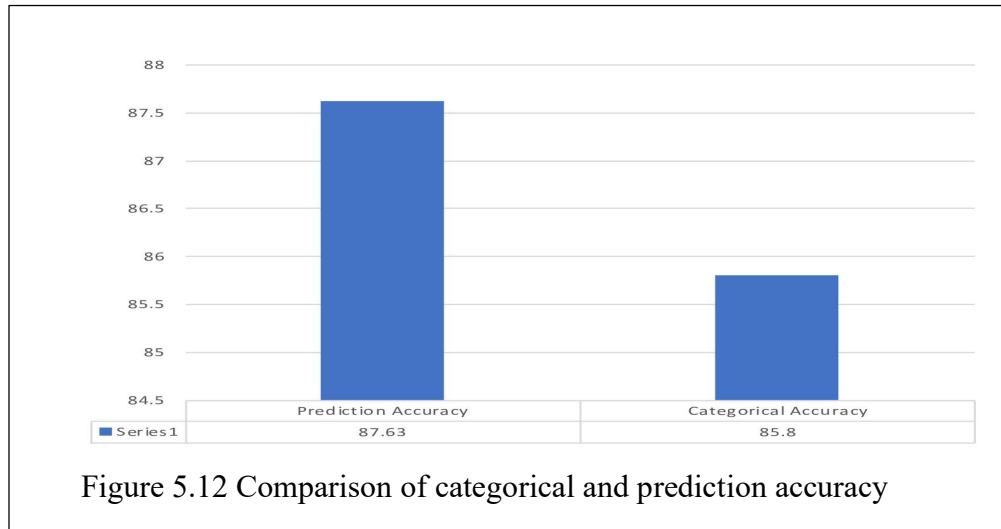
5.1.7 DEAL framework implementation on LED generator data set

The LED generator is a synthetic dataset. This simple domain contains seven Boolean attributes and 10 classes, the set of decimal digits. The class attribute is an integer ranging between 0 and 9, representing the possible digits shown on display [7]. For experimentation purpose, 10, 000 instances were generated, and the training and testing ratio was 80:20. Table 5.8 shows the evaluation results of the research algorithm DEAL on the LED data set.

Table 5.8 Results of DEAL over LED generator data set

S. No.	Evaluation parameter	Result
1	Prediction accuracy	87.63
2	F1-score	0.86
3	Recall	0.93
4	Precision	0.83
5	Categorical accuracy	85.8

The graph in Figure 5.12 compares the prediction accuracy and categorical accuracy of DEAL over the LED generator data set.



5.1.8 DEAL framework implementation on RBF data set

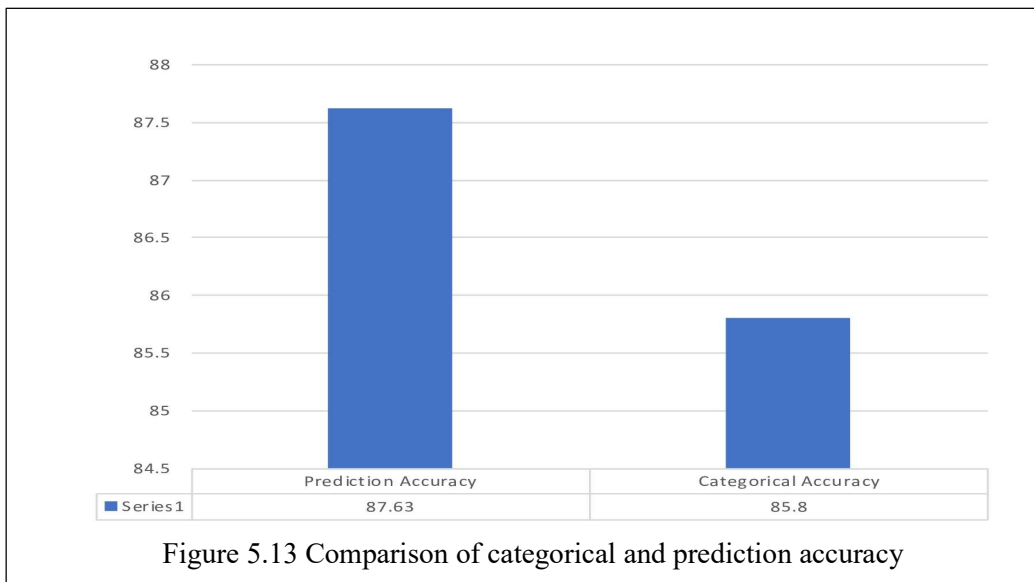
The RBF dataset is a synthetic dataset. The random RBF generator has 10 dimensions and five classes [7]. For experimentation purpose, 10,000 instances of dataset were generated. The training and testing dataset were in the ratio 80:20.

Table 5.9 shows the evaluation results of the research algorithm DEAL on the RBF data set.

Table 5.9 Results of DEAL over RBF dataset

S. No.	Evaluation parameter	Result
1	Prediction accuracy	98.63
2	F1-score	0.87
3	Recall	0.91
4	Precision	0.92
5	Categorical accuracy	92.3

The graph in Figure 5.13 compares the prediction accuracy and categorical accuracy of DEAL over the RBF data set.



This section discussed the implementation results of the DEAL framework for binary and multiclass datasets. In each case, the prediction accuracy and categorical

accuracy has been compared. The comparisons indicate that the DEAL model has a good prediction accuracy and categorical accuracy value.

5.2 Conclusion

In this chapter, the DEAL framework implementation for data stream classification is discussed in detail. The experimentation was carried out using both real time and synthetic datasets for both binary and multiclass data streams. The datasets were varied, like CCFD dataset were highly imbalanced, stock prediction utilized special techniques for extracting features during learning phase as the available features were not enough to make predictions and the HAR dataset had overlapping features. The results with each dataset have been summarized and the prediction accuracy and categorical accuracy of DEAL framework for each dataset has also been presented. The previous works evaluated the classifiers only on basis of prediction accuracy and ignored categorical accuracy. As categorical accuracy is a significant metrics while evaluating data streams classifiers especially in imbalanced scenario, thus, good value of categorical accuracy proves the effectiveness of the DEAL framework for data stream classification. The performance of the DEAL framework on the binary datasets i.e., CCFD, stock prediction, SEA generator and hyperplane framework in terms of accuracy is 99.81,66.61,93.76 and 92.56 respectively. While, the performance in terms of categorical accuracy is 99.79,59.2,90.03 and 89.7 respectively. The performance of the DEAL framework on the multiclass datasets i.e., HAR, Poker hand, RBF and LED generator in terms of accuracy is 98.7, 99.67, 98.63 and 87.63 respectively. While, the performance in terms of categorical accuracy is 83.18, 97.3, 92.3 and 85.8 respectively. In the next chapter, the DEAL framework is compared with recent and benchmark algorithms. Statistical analysis has been done to further verify the obtained results.

Chapter 6

Performance Comparison of DEAL Framework and Statistical Analysis

This chapter presents the performance comparison details of the DEAL framework with the state-of-the-art algorithms and the standard works. Further, the chapter also details the statistical results to further verify the efficiency of the proposed DEAL framework. The Chapter-5 has presented the experiment results of DEAL framework, and the compared algorithms were also experimented on the same experimental platform. The results thus obtained were compared with the results of the DEAL framework. The evaluation metrics used in the comparison are precision, recall, accuracy, F1-score and categorical accuracy. Section 6.1 presents the comparison with benchmark algorithms, section 6.2 presents the comparison with state-of-art algorithms, and Section 6.3 presents the statistical analysis.

6.1 Comparison with benchmark algorithms

In this section, the performance of the DEAL framework is compared with benchmark algorithms like SVM[104], AE, CNN[104], LR[43] and MLP[105], over same binary and multiclass datasets used for implementing DEAL framework in chapter 5. Table 6.1 summarizes the implementation results of the DEAL framework and benchmark algorithms over eight datasets and Table 6.2 presents the average performance of DEAL and benchmark algorithms.

Table 6.1 Implementation results of the DEAL framework and benchmark algorithms

Data set	Classifiers	Accuracy	Categorical accuracy	F1-score	Precision	Recall
CCFD	DEAL	99.81	99.79	0.99	0.99	0.99
	SVM	99.94	67	0.66	0.612	0.6
	LR	99.91	63	0.5	0.54	0.56
	CNN	99.89	82	0.732	0.71	0.683
	MLP	99.94	81	0.61	0.65	0.62
	AE	96.03	83.67	0.67	0.72	0.59
Stock prediction	DEAL	0.6661	0.592	0.516	0.45	0.69
	SVM	51.13	38.6	0.47	0.41	0.621
	LR	51.6	43.1	0.49	0.38	0.56
	CNN	58.93	42.7	0.44	0.32	0.683
	MLP	61.32	50.8	0.5	0.38	0.62
	AE	59.8	46.71	0.34	0.34	0.59
HAR	DEAL	0.9872	0.8318	0.989	0.98	0.99
	SVM	0.875	0.6314	0.876	0.882	0.875
	LR	0.8076	0.5978	0.811	0.853	0.807
	CNN	0.905	0.713	0.876	0.882	0.875
	MLP	0.9519	0.7328	0.952	0.9555	0.9519
	AE	0.6346	0.5672	0.735	0.834	0.6346
Poker hand	DEAL	0.9967	0.973	0.74	0.82	0.65
	SVM	0.748	0.543	0.66	0.725	0.52
	LR	0.909	0.641	0.53	0.56	0.46
	CNN	0.9671	0.727	0.54	0.71	0.63
	MLP	0.9541	0.808	0.61	0.66	0.52
	AE	0.9256	0.7671	0.68	0.64	0.49

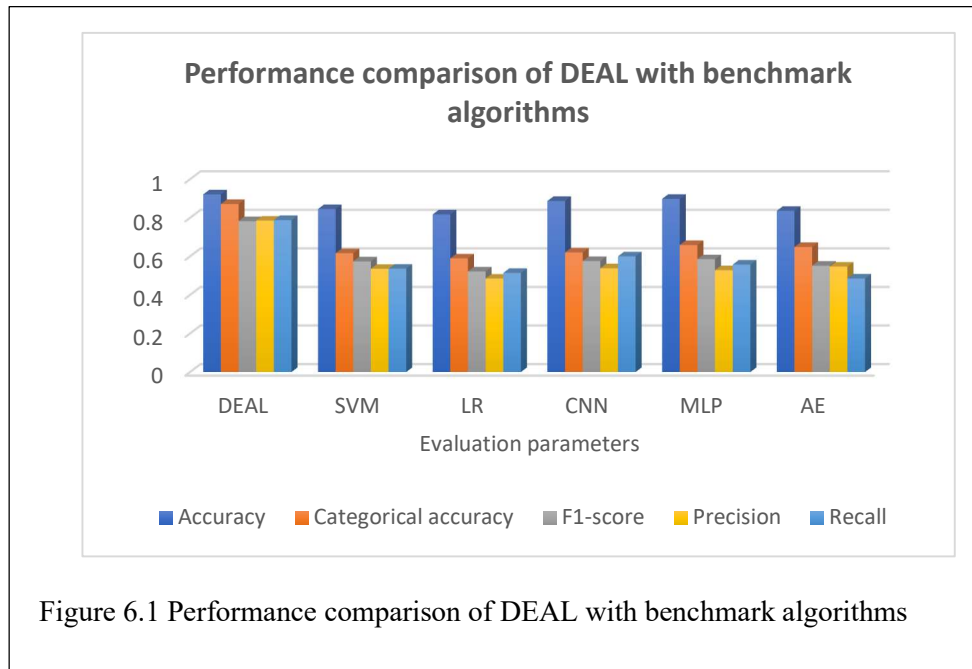
Table 6.1 continues..

Data set	Classifiers	Accuracy	Categorical accuracy	F1-score	Precision	Recall
Hyperplane	DEAL	0.9256	0.897	0.79725	0.8	0.81275
	SVM	0.9178	0.6378	0.547	0.419	0.414
	LR	0.8372	0.6072	0.413	0.339	0.465
	CNN	0.9168	0.4768	0.451	0.447	0.476
	MLP	0.9234	0.591	0.503	0.345	0.484
	AE	0.8978	0.6061	0.472	0.436	0.327
LED	DEAL	0.8763	0.858	0.73231	0.738	0.75694
	SVM	0.899	0.6138	0.422	0.449	0.436
	LR	0.789	0.6363	0.592	0.414	0.368
	CNN	0.9073	0.6271	0.578	0.345	0.462
	MLP	0.9362	0.6312	0.454	0.327	0.339
	AE	0.8902	0.6532	0.584	0.476	0.318
RBF	DEAL	0.9863	0.923	0.79464	0.81075	0.76992
	SVM	0.8868	0.7128	0.537	0.349	0.337
	LR	0.8568	0.5768	0.357	0.352	0.377
	CNN	0.9132	0.5578	0.406	0.449	0.46
	MLP	0.8904	0.589	0.582	0.462	0.461
	AE	0.878	0.672	0.434	0.465	0.463
SEA	DEAL	0.9376	0.9003	0.70349	0.69694	0.64771
	SVM	0.9263	0.7362	0.421	0.437	0.488
	LR	0.8264	0.5962	0.481	0.44	0.516
	CNN	0.9008	0.612	0.584	0.442	0.533
	MLP	0.918	0.6013	0.472	0.447	0.463
	AE	0.907	0.6208	0.497	0.466	0.467

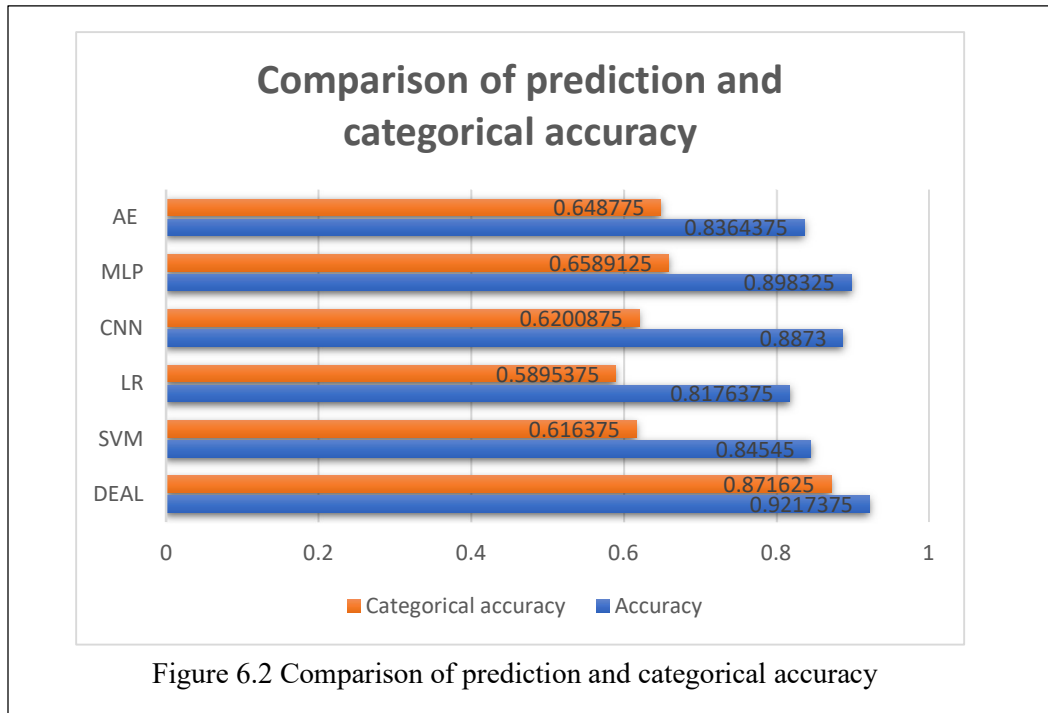
Table 6.2 Average performance of DEAL and benchmark algorithms

Classifiers	Accuracy	Categorical accuracy	F1-score	Precision	Recall
DEAL	0.921738	0.871625	0.841875	0.84125	0.845
Self-paced	0.891257	0.659563	0.699875	0.7325	0.68125
Hybrid	0.81485	0.630463	0.626125	0.69375	0.67875
Boosting-based ensemble	0.715888	0.676163	0.686375	0.75875	0.70625

Graph in Figure 6.1 compares the performance of the DEAL framework with the recent algorithms over all eight experimental datasets.



In figure 6.2, prediction accuracy and categorical accuracies are compared



approximately 24% better than the compared benchmark algorithms, Thus, the DEAL framework is superior than the compared recent algorithms in accuracy as well as categorical accuracy.

The next section, compares the performance of DEAL framework with state-of-art algorithms.

6.2 Comparison with State-of-the-art Algorithms

In this section, the performance of the DEAL framework is compared with recent algorithms like self-paced[96], hybrid algorithm [98], and a iterative boosting based learning algorithm[97], over same binary and multiclass datasets used for implementing DEAL framework in chapter 5. Table 6.3 summarizes the implementation results of the DEAL framework and three recent algorithms over eight datasets and Table 6.4 presents the average performance of DEAL and stat

Table 6.3 Implementation results of the DEAL framework and three recent algorithms

Data set	Classifiers	Accuracy	Categorical accuracy	F1-score	Precision	Recall
CCFD	DEAL	0.9981	0.9979	0.99	0.99	0.99
	Self-paced	0.9765	0.7618	0.89	0.89	0.89
	Hybrid	0.8338	0.658	0.52	0.66	0.76
	Boosting-based ensemble	0.9672	0.7683	0.78	0.79	0.69
Stock exchange	DEAL	0.6661	0.592	0.516	0.45	0.69
	Self-paced	0.5992	0.4817	0.491	0.44	0.54
	Hybrid	0.6083	0.4677	0.43	0.36	0.67
	Boosting-based ensemble	0.5371	0.5123	0.397	0.48	0.59
Hyperplane	DEAL	0.9256	0.897	0.91	0.87	0.79
	Self-paced	0.8996	0.626	0.83	0.79	0.63
	Hybrid	0.9012	0.689	0.76	0.68	0.51
	Boosting-based ensemble	0.8984	0.654	0.81	0.79	0.69
SEA	DEAL	0.9376	0.9003	0.86	0.87	0.81
	Self-paced	0.8992	0.723	0.64	0.75	0.74
	Hybrid	0.8992	0.698	0.78	0.82	0.78
	Boosting-based ensemble	0.89.8	0.745	0.72	0.74	0.76
HAR	DEAL	0.9872	0.8318	0.989	0.98	0.99
	Self-paced	0.9298	0.6319	0.89	0.91	0.85
	Hybrid	0.8071	0.673	0.54	0.76	0.87
	Boosting-based ensemble	0.8407	0.6543	0.86	0.88	0.78
Poker hand	DEAL	0.9967	0.973	0.74	0.82	0.65
	Self-paced	0.9925	0.7241	0.438	0.73	0.49
	Hybrid	0.786	0.5609	0.519	0.64	0.51
	Boosting-based ensemble	0.8012	0.7234	0.344	0.79	0.51
LED	DEAL	0.8763	0.858	0.86	0.83	0.93
	Self-paced	0.7444	0.589	0.74	0.64	0.66
	Hybrid	0.7412	0.537	0.78	0.76	0.61
	Boosting-based ensemble	0.7424	0.554	0.8	0.77	0.84
RBF	DEAL	0.9863	0.923	0.87	0.92	0.91
	Self-paced	0.942	0.739	0.68	0.71	0.65
	Hybrid	0.942	0.7601	0.68	0.87	0.72
	Boosting-based ensemble	0.9401	0.798	0.78	0.83	0.79

Table 6.4 Average performance of DEAL and recent algorithms

Classifiers	Accuracy	Categorical accuracy	F1-score	Precision	Recall
DEAL	0.921738	0.871625	0.841875	0.84125	0.845
Self-paced	0.891257	0.659563	0.699875	0.7325	0.68125
Hybrid	0.81485	0.630463	0.626125	0.69375	0.67875
Boosting-based ensemble	0.715888	0.676163	0.686375	0.75875	0.70625

Graph in Figure 6.3 compares the performance of the DEAL framework with the recent algorithms over all experimental datasets.

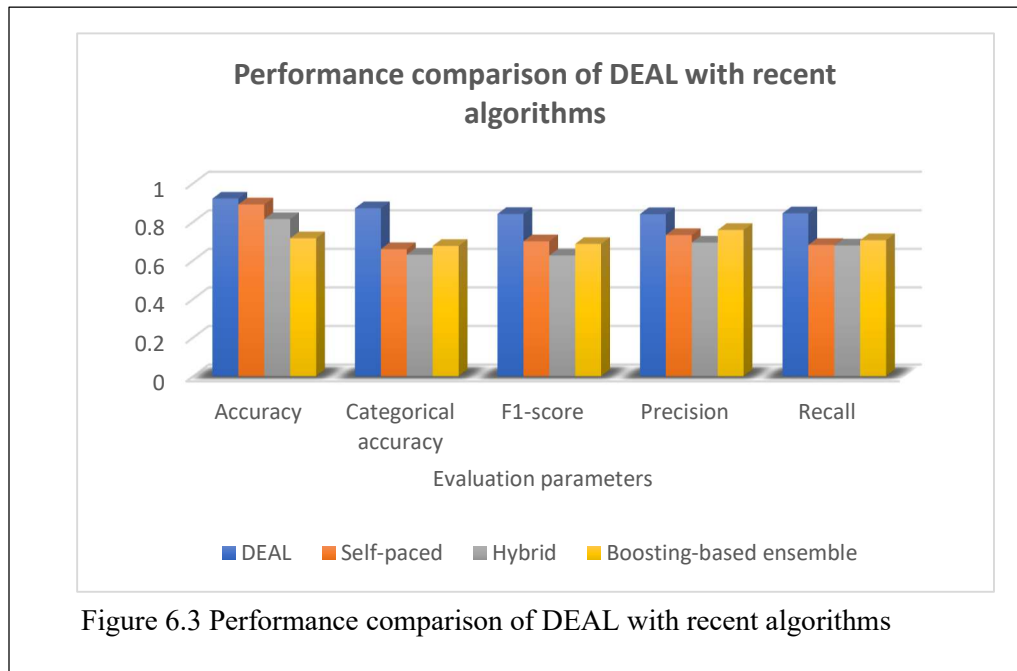
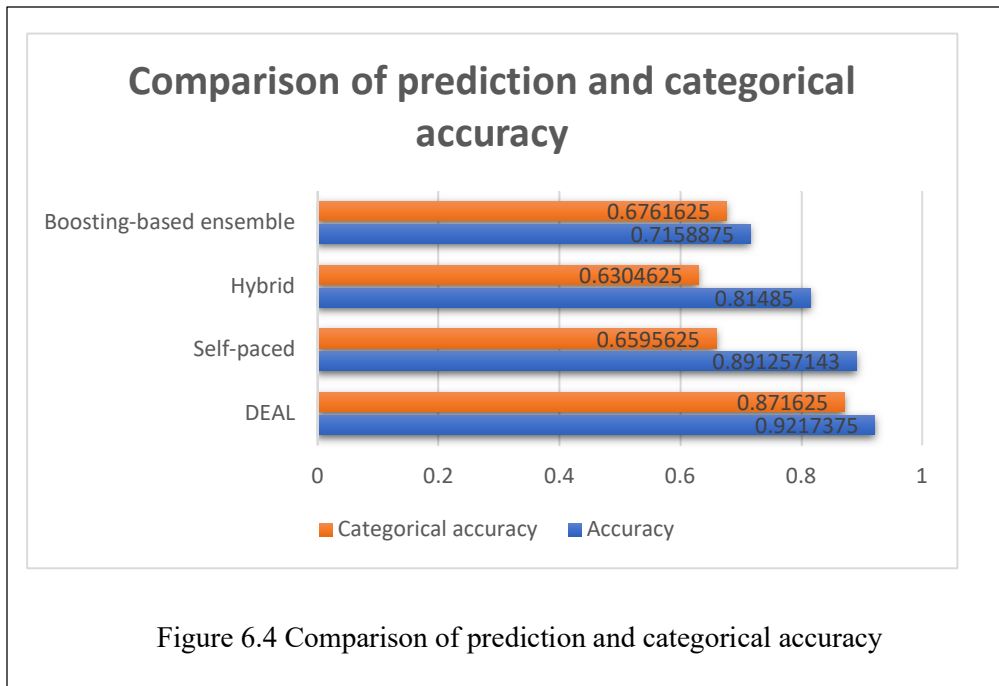


Figure 6.3 Performance comparison of DEAL with recent algorithms

In figure 6.4, prediction accuracy and categorical accuracies are compared.



The comparison in figure 6.2 the categorical accuracy of DEAL is approximately 21% better than the compared recent algorithms, Thus, the DEAL framework is superior than the compared recent algorithms in accuracy as well as categorical accuracy.

In section 6.1 and 6.2, the performance of the proposed DEAL framework is compared with the benchmark and state-of-art algorithms. The results show that the proposed model performs better than the compared works in terms of prediction accuracy as well as categorical accuracy. The categorical accuracy improves by approximately 22.5 %. In addition, the values of the other evaluation metrics are also good. Further, the results are verified using Statistical analysis.

6.3 Statistical analysis for data stream classification

Statistical analysis enables the evaluation of assertions using quantitative evidence. This analysis assists in distinguishing between reasonable and doubtful findings. When analysts employ appropriate statistical approaches, they frequently produce accurate results. Indeed, statistical methods incorporate uncertainty and error into their conclusions. In this work, Wilcoxon signed-rank test[106] is used for statistical analysis. This test compares two related samples, matched samples, or performs a paired difference test of repeated measurements on a single sample to see if their population mean ranks differ. The Wilcoxon signed-rank test is a statistical test. It is frequently used in hypothesis testing to determine whether a process or treatment has an actual effect on the population of interest or whether two groups are significantly different from one another, among other things. Wilcoxon signed-rank test is a mathematical procedure that takes a sample from each set and establishes the problem statement by assuming the null hypothesis that the two means are equal. Then, specific values are calculated and compared against the standard values using the applicable formulas. The assumed null hypothesis is either accepted or rejected based on the results of the comparison. Whenever the null hypothesis meets the criteria for being rejected, it indicates that the data readings are strong and are likely not the result of random chance. In this work, a Wilcoxon signed-rank test is applied to test whether the proposed framework for classifying data streams has significantly improved performance than conventional and related works. Accordingly, a hypothesis is designed to compare the performance of the proposed work with traditional and recent works based on accuracy and categorical accuracy.

6.3.1 Statistical analysis for accuracy

6.3.1.1 Self-paced Accuracy Vs DEAL Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
Self - paced	8	0	8	59.920	99.250	87.290	13.410
DEAL	8	0	8	66.610	99.810	92.174	11.186

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The prediction accuracy of DEAL and self-paced algorithm is similar.

Ha: The prediction accuracy of DEAL is improved in comparison with self-paced algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that accuracy of DEAL and self-paced algorithm is not same and the accuracy of DEAL algorithm is improved in comparison with self-paced algorithm.

6.3.1.2 Hybrid Accuracy Vs DEAL Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
hybrid	8	0	8	60.830	94.200	81.485	10.696
DEAL	8	0	8	66.610	99.810	92.174	11.186

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H₀: The prediction accuracy of DEAL and hybrid algorithm is similar.

H_a: The prediction accuracy of DEAL is improved in comparison with hybrid algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H₀, and accept the alternative hypothesis H_a.

Conclusion:

The acceptance of alternate hypothesis H_a indicates that accuracy of DEAL and Hybrid algorithm is not same and the accuracy DEAL algorithm is improved in comparison with Hybrid algorithm.

6.3.1.3 Boosting Accuracy Vs DEAL Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
Boosting	8	0	8	53.710	96.720	82.787	13.853
DEAL	8	0	8	66.610	99.810	92.174	11.186

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The prediction accuracy of DEAL and boosting algorithm is similar.

Ha: The prediction accuracy of DEAL is improved in comparison with boosting algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that accuracy of DEAL and boosting algorithm is not same and the accuracy of DEAL algorithm is improved in comparison with boosting algorithm.

6.3.1.4 SVM Accuracy Vs DEAL Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
SVM	8	0	8	51.130	99.940	84.545	15.216
DEAL	8	0	8	66.610	99.810	92.174	11.186

Sign test / Lower-tailed test:

N+	2
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.145
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	5
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.039
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H₀: The prediction accuracy of DEAL and SVM algorithm is similar.

H_a: The prediction accuracy of DEAL is improved in comparison with SVM algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H₀, and accept the alternative hypothesis H_a.

Conclusion:

The acceptance of alternate hypothesis H_a indicates that accuracy of DEAL and SVM algorithm is not same and the accuracy of DEAL algorithm is improved in comparison with SVM algorithm.

6.3.1.5 LR Accuracy Vs DEAL Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
LR	8	0	8	51.600	99.910	81.764	13.896
DEAL	8	0	8	66.610	99.810	92.174	11.186

Sign test / Lower-tailed test:

N+	1
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.035
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	1
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.008
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The prediction accuracy of DEAL and LR is similar.

Ha: The prediction accuracy of DEAL is improved in comparison with LR.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that accuracy of DEAL and LR algorithm is not same and the accuracy of the DEAL algorithm is improved in comparison with LR algorithm.

6.3.1.6 MLP Accuracy Vs DEAL Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
MLP	8	0	8	61.320	99.940	89.833	11.953
DEAL	8	0	8	66.610	99.810	92.174	11.186

Sign test / Lower-tailed test:

N+	2
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.145
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	8
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.098
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H₀: The prediction accuracy of DEAL and MLP is similar.

H_a: The prediction accuracy of DEAL is improved in comparison with MLP.

As the computed p-value is greater than the significance level $\alpha=0.05$, one cannot reject the null hypothesis H₀.

Conclusion:

The acceptance of alternate hypothesis H_a indicates that accuracy of DEAL and MLP algorithm is not same and the accuracy of the DEAL algorithm is improved in comparison with MLP algorithm.

6.3.1.7 CNN Accuracy Vs DEAL Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
CNN	8	0	8	58.930	99.890	88.730	12.539
DEAL	8	0	8	66.610	99.810	92.174	11.186

Sign test / Lower-tailed test:

N+	2
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.145
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	5
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.039
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The prediction accuracy of DEAL and CNN is similar.

Ha: The prediction accuracy of DEAL is improved in comparison with CNN.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that accuracy of DEAL and CNN is not same and the accuracy of the DEAL algorithm is improved in comparison with CNN.

6.3.1.8 AE Accuracy Vs DEAL Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
AE	8	0	8	59.800	96.030	83.644	13.848
DEAL	8	0	8	66.610	99.810	92.174	11.186

Sign test / Lower-tailed test:

N+	1
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.035
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	1
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.008
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H₀: The prediction accuracy of DEAL and AE is similar.

H_a: The prediction accuracy of DEAL is improved in comparison with AE.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H₀, and accept the alternative hypothesis H_a.

Conclusion:

The acceptance of alternate hypothesis H_a indicates that accuracy of DEAL and AE is not same and the accuracy of the DEAL algorithm is improved in comparison with AE.

6.3.2 Statistical analysis for categorical accuracy

6.3.2.1 Self-paced Categorical Accuracy Vs DEAL Categorical Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
Self - paced	8	0	8	48.170	76.180	65.956	9.521
DEAL	8	0	8	59.200	99.790	87.163	12.552

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The categorical accuracy of DEAL and self-paced algorithm is similar.

Ha: The categorical accuracy of DEAL is improved in comparison with self-paced algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that categorical accuracy of DEAL and self-paced algorithm is not same and the categorical accuracy of the DEAL algorithm is improved in comparison with self-paced algorithm.

6.3.2.2 Hybrid Categorical Accuracy Vs DEAL Categorical Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
hybrid	8	0	8	46.770	76.010	63.046	9.815
DEAL	8	0	8	59.200	99.790	87.163	12.552

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The categorical accuracy of DEAL and hybrid algorithm is similar.

Ha: The categorical accuracy of DEAL is improved in comparison with hybrid algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis H_a indicates that categorical accuracy of DEAL and hybrid algorithm is not same and the categorical accuracy of the DEAL algorithm is improved in comparison with hybrid algorithm.

6.3.2.3 Boosting Categorical Accuracy Vs DEAL Categorical Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
Boosting	8	0	8	51.230	79.800	67.616	10.215
DEAL	8	0	8	59.200	99.790	87.163	12.552

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The categorical accuracy of DEAL and boosting algorithm is similar.

Ha: The categorical accuracy of DEAL is improved in comparison with boosting algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that categorical accuracy of DEAL and Boosting algorithm is not same and the categorical accuracy of the DEAL algorithm is improved in comparison with Boosting algorithm.

6.3.2.4 SVM Categorical Accuracy Vs DEAL Categorical Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
SVM	8	0	8	38.600	73.620	61.638	11.063
DEAL	8	0	8	59.200	99.790	87.163	12.552

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The categorical accuracy of DEAL and SVM is similar.

Ha: The categorical accuracy of DEAL is improved in comparison with SVM algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that categorical accuracy of DEAL and SVM is not same and the categorical accuracy of the DEAL algorithm is improved in comparison with SVM.

6.3.2.5 DEAL Categorical Accuracy VS LR Categorical Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
LR	8	0	8	43.100	64.100	58.954	6.782
DEAL	8	0	8	59.200	99.790	87.163	12.552

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The categorical accuracy of DEAL and LR is similar.

Ha: The categorical accuracy of DEAL is improved in comparison with LR algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis H_a indicates that the categorical accuracy of DEAL and LR is not same and the categorical accuracy of the DEAL algorithm is improved in comparison with LR.

6.3.2.6 AE Categorical Accuracy Vs DEAL Categorical Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
AE	8	0	8	46.710	83.670	64.878	11.473
DEAL	8	0	8	59.200	99.790	87.163	12.552

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The categorical accuracy of DEAL and AE is similar.

Ha: The categorical accuracy of DEAL is improved in comparison with AE algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that the categorical accuracy of DEAL and AE is not same and the categorical accuracy of the DEAL algorithm is improved in comparison with AE.

6.3.2.7 MLP Categorical Accuracy Vs DEAL Categorical Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
MLP	8	0	8	50.800	81.000	65.891	11.140
DEAL	8	0	8	59.200	99.790	87.163	12.552

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The categorical accuracy of DEAL and MLP is similar.

Ha: The categorical accuracy of DEAL is improved in comparison with MLP algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis Ha indicates that the categorical accuracy of DEAL and MLP is not same and the categorical accuracy of the DEAL algorithm is improved in comparison with MLP.

6.3.2.8 CNN Categorical Accuracy Vs DEAL Categorical Accuracy

Summary statistics:

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
CNN	8	0	8	42.700	82.000	62.009	13.193
DEAL	8	0	8	59.200	99.790	87.163	12.552

Sign test / Lower-tailed test:

N+	0
Expected value	4.000
Variance (N+)	2.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method.

Wilcoxon signed-rank test / Lower-tailed test:

V	0
Expected value	18.000
Variance (V)	51.000
p-value (one-tailed)	0.004
alpha	0.050

The p-value is computed using an exact method. Time elapsed: 0s.

Test interpretation:

H0: The categorical accuracy of DEAL and CNN is similar.

Ha: The categorical accuracy of DEAL is improved in comparison with CNN algorithm.

As the computed p-value is lower than the significance level $\alpha=0.05$, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.

Conclusion:

The acceptance of alternate hypothesis H_a indicates that the categorical accuracy of DEAL and CNN is not same and the categorical accuracy of the DEAL algorithm is improved in comparison with CNN.

6.4 Conclusion

The performance comparison of DEAL with recent and benchmark algorithms has been discussed in this chapter. The comparison shows that the DEAL framework has overall improved performance over compared algorithms in accuracy and categorical accuracy. The other evaluation metrics measures are also good. Finally, Wilcoxon signed-rank test has been performed to test the statistical significance of the proposed DEAL algorithm. The acceptance of the alternate hypothesis indicates that the proposed framework is efficient in classifying the data streams. In chapter 7, the conclusion and future research directions of the present research program are presented.

Chapter 7

Conclusion and Future Research Directions

7.1 Summary

Traditional ML algorithms have proven to be beneficial in extracting knowledge from static data. However, they are not suited for dynamic data stream environments. This raises the need of using an algorithm that can obtain data streams from heterogeneous sources and synthesize these diverse data streams into a single source of knowledge to make smart decisions. As such, the tasks that data stream classifiers must perform are becoming increasingly complicated and multifaceted. In a similar manner to how the human brain combines the five senses to understand what is truly occurring in real time, future data stream classification algorithms are expected to interpret high-speed data streams. To achieve seamless performance that can mimic the human brain, proper integration of data ingestion, synthesis, and timely action is needed when dealing with such massive amounts of data from data streams.

The technology to underpin these systems is already emerging. The application of advanced algorithms such as deep learning (DL), artificial intelligence (AI), and optimization in real-time Big Data is gaining significant interest due to their ability to discover and predict unseen patterns. The application of advanced algorithms and optimizations is not limited to a single area but can be applied to tackle various challenges associated with any domain. Some key areas are sensor networks, health monitoring, networks, finance, power, security, and privacy. AI and advanced ML

techniques and optimization will certainly enhance discovery and experience by addressing various challenges and issues.

This research emphasizes developing and discovering advanced algorithms and optimizations for data stream predictions and related paradigms. The objective of the present research program is to “Design and implement an efficient ensemble-based classification algorithm for real-time data streams.” A detailed introduction to the data stream and data stream classification is provided at the outset of the thesis. Next, the importance of data stream classification is discussed. Furthermore, the research objective of the current research program is presented in that chapter as well. Chapter 2 is devoted to a thorough review of the literature on techniques for data stream classification and methods for improving the performance of data stream classifiers. The survey reveals that some techniques perform better in binary classification while others perform better in multiclass classification scenarios.

Furthermore, we discover that no technique in the literature performs well for binary and multiclass classification while preventing the model from overfitting at the same time. The evaluation metrics used are insufficient for the data stream environment. In Chapter 3, the proposed framework is described in detail. Chapter 4 presents the devised algorithms and the experimental environment to implement the proposed work. Chapters 5 discuss the implementation results in binary and multiclass scenarios. Chapters 6 compare the proposed work with some recent and benchmark algorithms. Chapter 7 presents the summary in Section 7.1, followed by the significant contribution of this research work in Section 7.2. Finally, Section 7.3 concludes the current research program with recommendations for future research.

7.2 Contributions

(Contribution 1): A novel deep learning based framework for Efficient Data Stream Classification

A framework for efficient classification of data streams is proposed. The proposed framework consists of two algorithms for learning/training the DL model and, finally, an algorithm that optimizes the DL model for improving the model's performance. The present research work has developed deep learning framework for efficient data stream classification. The framework is capable of classifying binary and multiclass data streams. The DEAL framework is evaluated based on categorical accuracy. Categorical accuracy is crucial for evaluating data stream models, especially when the stream is unbalanced. The results are further verified using statistical analysis.

(Contribution 2): Extra Tree Feature Ensemble-Based Optimization to overcome Concept Drift in data streams

The research work has proposed the extra tree feature ensemble approach for optimization. The approach incorporates the newly arriving feature due to concept drift in the feature subset and trains the model. This approach overcomes the concept drift issue in data streams. Optimization also removes irrelevant features from the feature set, and thus, the overfitting problem is also avoided in the proposed work.

7.3 Limitations and Future Directions

Data stream classification is one of the extremely important areas of data mining. The current study can be expanded in numerous directions to further enhance this area. One of the limitations of the present work is that, the DEAL

model can become costly with the increase in dimensions of data streams. In future, the kernel trick can be applied to provide a more efficient and cost-effective method of transforming data into higher dimensions. As all the data is not linearly separable in the actual world and almost all data is randomly distributed, it is difficult to categorize them linearly. When the number of dimensions increases, computations within that space become increasingly expensive. The kernel trick in such a scenario enables the system to work in the original feature space without having to compute the data's coordinates in a higher dimensional space, thus making the system cost effective. Data stream classification is still in the early stages of development to meet its intended purpose, and significant research is ongoing. This work opens up several co-fronts on forecasting patterns and classification on other real-time transaction data. The present work has less consideration for data privacy. In future, to ensure data protection and real-time analysis, the triumvirate (blockchain, internet of things (IoT), and AI) can be utilized in cloud computing platforms to solve the complex problems of next-generation computing. Blockchain can improve security by decentralizing data in software-defined networks. AI can further detect and predict failures and improve fault tolerance. IoT can enable serverless computing, improve future systems, and help in designing more innovative applications. The proposed DEAL model can also be further optimized using several bio-inspired algorithms. Big Data acquisition framework and technologies like Kafka and Flume can be integrated with the proposed work to develop automated models for real-time applications. The present work demands high memory requirements. To overcome this constraint and to facilitate utilization in low-end memory devices, a distributed processing version of the proposed model can be developed in future.

References

- [1] M. Friendly, “Milestones in the history of thematic cartography , statistical graphics , and data visualization,” *Engineering*, vol. 9, p. 2008, 2009, doi: 10.1016/S1360-1385(01)02193-8.
- [2] K. Dick Basu A S, “from the SAGE Social Science Collections . All Rights,” *Hisp. J. Behav. Sci.*, vol. 9, no. 2, pp. 183–205, 1987, [Online]. Available: <http://hjb.sagepub.com.proxy.lib.umich.edu/content/9/2/183.full.pdf+html>.
- [3] L. Peltason and J. Bajorath, “Systematic computational analysis of structure-activity relationships: Concepts, challenges and recent advances,” *Future Med. Chem.*, vol. 1, no. 3, pp. 451–466, 2009, doi: 10.4155/fmc.09.41.
- [4] A. L. Lederer and A. L. Mendelow, “Issues in information systems planning,” *Inf. Manag.*, vol. 10, no. 5, pp. 245–254, 1986, doi: 10.1016/0378-7206(86)90027-3.
- [5] G. Halevi and H. Moed, “The Evolution of Big Data as a Research and Scientific Topic: Overview of the Literature,” *Res. Trends*, vol. 30, no. 36, pp. 3–6, 2012.
- [6] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “Data Mining and Knowledge Discovery Handbook,” *Data Min. Knowl. Discov. Handb.*, 2010, doi: 10.1007/978-0-387-09823-4.
- [7] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems,” p. 1, 2002, doi: 10.1145/543614.543615.
- [8] F. Opitz, K. Dastner, B. V. H. Z. Roseneckh-Kohler, and E. Schmid, “Data analytics and machine learning in wide area surveillance systems,” *Proc. Int. Radar Symp.*, vol. 2019-June, no. 1, pp. 1–10, 2019, doi:

10.23919/IRS.2019.8768102.

- [9] E. Alothali, H. Alashwal, and S. Harous, “Data stream mining techniques: A review,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 2, pp. 728–737, 2019, doi: 10.12928/TELKOMNIKA.v17i2.11752.
- [10] S. Gupta, “A Regression Modeling Technique on Data Mining,” *Int. J. Comput. Appl.*, vol. 116, no. 9, pp. 27–29, 2015, doi: 10.5120/20365-2570.
- [11] A. Marradi, “Classification, typology, taxonomy,” *Qual. Quant.*, vol. 24, no. 2, pp. 129–157, 1990, doi: 10.1007/BF00209548.
- [12] O. Rusu *et al.*, “Converting unstructured and semi-structured data into knowledge,” *Proc. - RoEduNet IEEE Int. Conf.*, pp. 1–4, 2013, doi: 10.1109/RoEduNet.2013.6511736.
- [13] G. D. F. Morales, A. Bifet, L. Khan, J. Gama, and W. Fan, “IoT big data stream mining,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Aug, pp. 2119–2120, 2016, doi: 10.1145/2939672.2945385.
- [14] J. E. Nalavade, “Challenges in data stream classification 1,” no. 10, pp. 27–35, 2015.
- [15] A. Bifet *et al.*, “MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering,” *J. Mach. Learn. Res. Work. Conf. Proc.*, vol. 11, pp. 44–50, 2010.
- [16] Z. Allah Bukhsh, A. Saeed, I. Stipanovic, and A. G. Doree, “Predictive maintenance using tree-based classification techniques: A case of railway switches,” *Transp. Res. Part C Emerg. Technol.*, vol. 101, no. January, pp. 35–54, 2019, doi: 10.1016/j.trc.2019.02.001.
- [17] X. Zhou, L. La, and F. Klawonn, “Evolving Fuzzy Rule-based Classifiers,”

no. Ciisp, pp. 220–225, 2007.

- [18] H. M. Gomes, J. P. Barddal, A. F. Enembreck, and A. Bifet, “A survey on ensemble learning for data stream classification,” *ACM Comput. Surv.*, vol. 50, no. 2, 2017, doi: 10.1145/3054925.
- [19] S. A. Babu, “Ensemble Technique for Efficient Stream Data Processing,” vol. 10, no. 5, pp. 1531–1542, 2017.
- [20] V. Bolón-Canedo and A. Alonso-Betanzos, “Ensembles for feature selection: A review and future trends,” *Inf. Fusion*, vol. 52, no. May 2018, pp. 1–12, 2019, doi: 10.1016/j.inffus.2018.11.008.
- [21] A. Ben Brahim and M. Limam, “Ensemble feature selection for high dimensional data: a new method and a comparative study,” *Adv. Data Anal. Classif.*, pp. 1–16, 2017, doi: 10.1007/s11634-017-0285-y.
- [22] A. Jurek, Y. Bi, S. Wu, and C. Nugent, “A survey of commonly used ensemble-based classification techniques,” *Knowl. Eng. Rev.*, vol. 29, no. 5, pp. 551–581, 2013, doi: 10.1017/S0269888913000155.
- [23] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, “Deep learning for IoT big data and streaming analytics: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018, doi: 10.1109/COMST.2018.2844341.
- [24] X. Jia, “Image recognition method based on deep learning,” *Proc. 29th Chinese Control Decis. Conf. CCDC 2017*, pp. 4730–4735, 2017, doi: 10.1109/CCDC.2017.7979332.
- [25] I. Colkesen and T. Kavzoglu, “Selection of Optimal Object Features in Object-Based Image Analysis Using Filter-Based Algorithms,” *J. Indian*

- Soc. Remote Sens.*, vol. 46, no. 8, pp. 1233–1242, 2018, doi: 10.1007/s12524-018-0807-x.
- [26] D. Mladenić, “Feature selection for dimensionality reduction,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3940 LNCS, pp. 84–102, 2006, doi: 10.1007/11752790_5.
- [27] S. S. Hameed, O. O. Petinrin, A. O. Hashi, and F. Saeed, “Filter-wrapper combination and embedded feature selection for gene expression data,” *Int. J. Adv. Soft Comput. its Appl.*, vol. 10, no. 1, pp. 90–105, 2018.
- [28] C. Zhang and J. Han, *Data Mining and Knowledge Discovery*. 2021.
- [29] L. Khan, “Data Stream Mining: Challenges and Techniques,” pp. 295–295, 2010, doi: 10.1109/ictai.2010.114.
- [30] G. Kreml et al., “Open challenges for data stream mining research,” *ACM SIGKDD Explor. Newsl.*, vol. 16, no. 1, pp. 1–10, 2014, doi: 10.1145/2674026.2674028.
- [31] N. Jiang and L. Gruenwald, “P14-Jiang.Pdf,” vol. 35, no. 1, 2006.
- [32] Y. Chi, H. Wane, and P. S. Yu, “Loadstar: Load shedding in data stream mining,” *VLDB 2005 - Proc. 31st Int. Conf. Very Large Data Bases*, vol. 3, pp. 1302–1305, 2005.
- [33] S. Mittal and S. Tyagi, “Performance evaluation of machine learning algorithms for credit card fraud detection,” *Proc. 9th Int. Conf. Cloud Comput. Data Sci. Eng. Conflu. 2019*, pp. 320–324, 2019, doi: 10.1109/CONFLUENCE.2019.8776925.
- [34] E. S. Sansano, *Machine learning-based techniques for indoor localization*

and human activity recognition through wearable devices . Machine learning-based techniques for indoor localization and human activity recognition through wearable devices ., no. November. 2020.

- [35] R. Samya and R. Rathipriya, “Predictive Analysis for Weather Prediction using Data Mining with ANN: A Study,” vol. 6, no. 2, pp. 149–153, 2016.
- [36] M. S. Abdul Razak and C. R. Nirmala, “A computing model for trend analysis in stock data stream classification,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 3, pp. 1602–1609, 2020, doi: 10.11591/ijeecs.v19.i3.pp1602-1609.
- [37] G. Dong, J. Han, P. S. Yu, L. V. S. Lakshmanan, J. Pei, and H. Wang, “Online Mining of Changes from Data Streams : Research Problems and Preliminary Results,” *ACM SIGMOD MPDS '03 San*, pp. 11–13, 2003.
- [38] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, “Credit card fraud detection using machine learning techniques: A comparative analysis,” *Proc. IEEE Int. Conf. Comput. Netw. Informatics, ICCNI 2017*, vol. 2017-Janua, pp. 1–9, 2017, doi: 10.1109/ICCNI.2017.8123782.
- [39] S. Rajora *et al.*, “A Comparative Study of Machine Learning Techniques for Credit Card Fraud Detection Based on Time Variance,” *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 1958–1963, 2019, doi: 10.1109/SSCI.2018.8628930.
- [40] K. Wong, *Scalable Machine Learning Techniques for Highly Imbalanced Credit Card Fraud Detection : A Comparative Study*. Springer International Publishing, 2018.
- [41] D. Brzezinski, “Prequential AUC : properties of the area under the ROC,” *Knowl. Inf. Syst.*, vol. 52, no. 2, pp. 531–562, 2017, doi: 10.1007/s10115-

017-1022-8.

- [42] F. Itoo and M. Satwinder, “A Comparative Analysis of Logistic Regression, Naïve Bayes, and KNN Machine Learning Algorithms for Credit Card Fraud Detection,” *Int. J. Inf. Technol.*, 2020, doi: 10.1007/s41870-020-00430-y.
- [43] C. Sudha and D. Akila, “Credit Card Fraud Detection Using AES Technique,” *Lect. Notes Networks Syst.*, vol. 118, pp. 91–98, 2020, doi: 10.1007/978-981-15-3284-9_11.
- [44] A. Subasi, K. Khateeb, T. Brahimi, and A. Sarirete, *Human activity recognition using machine learning methods in a smart healthcare environment*. Elsevier Inc., 2019.
- [45] L. B. Marinho, A. H. de Souza Junior, and P. P. R. Filho, “A new approach to human activity recognition using machine learning techniques,” *Adv. Intell. Syst. Comput.*, vol. 557, pp. 529–538, 2017, doi: 10.1007/978-3-319-53480-0_52.
- [46] S. Oniga and J. Süto, “Human activity recognition using neural networks,” *Proc. 2014 15th Int. Carpathian Control Conf. ICCCC 2014*, pp. 403–406, 2014, doi: 10.1109/CarpathianCC.2014.6843636.
- [47] M. Khannouz and T. Glatard, “A benchmark of data stream classification for human activity recognition on connected objects,” *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–17, 2020, doi: 10.3390/s20226486.
- [48] G. A. Oguntala *et al.*, “SmartWall: Novel RFID-Enabled Ambient Human Activity Recognition Using Machine Learning for Unobtrusive Health Monitoring,” *IEEE Access*, vol. 7, pp. 68022–68033, 2019, doi: 10.1109/ACCESS.2019.2917125.

- [49] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 259–268, 2015, doi: 10.1016/j.eswa.2014.07.040.
- [50] I. Kumar, K. Dogra, C. Utreja, and P. Yadav, "A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction," *Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2018*, no. Icicct, pp. 1003–1007, 2018, doi: 10.1109/ICICCT.2018.8473214.
- [51] X. Zhong and D. Enke, "Predicting the daily return direction of the stock market using hybrid machine learning algorithms," *Financ. Innov.*, vol. 5, no. 1, 2019, doi: 10.1186/s40854-019-0138-0.
- [52] N. Sirimevan, I. G. U. H. Mamalgaha, C. Jayasekara, Y. S. Mayuran, and C. Jayawardena, "Stock Market Prediction Using Machine Learning Techniques," *2019 Int. Conf. Adv. Comput. ICAC 2019*, pp. 192–197, 2019, doi: 10.1109/ICAC49085.2019.9103381.
- [53] H. M. Gomes *et al.*, "Adaptive random forests for evolving data stream classification," *Mach. Learn.*, vol. 106, no. 9–10, pp. 1469–1495, 2017, doi: 10.1007/s10994-017-5642-8.
- [54] S. Xu and J. Wang, "Dynamic extreme learning machine for data stream classification," *Neurocomputing*, vol. 238, pp. 433–449, 2017, doi: 10.1016/j.neucom.2016.12.078.
- [55] P. Ksieniewicz, M. Woźniak, B. Cyganek, A. Kasprzak, and K. Walkowiak, "Data stream classification using active learned neural networks," *Neurocomputing*, vol. 353, pp. 74–82, 2019, doi: 10.1016/j.neucom.2018.05.130.

- [56] D. Mena-Torres and J. S. Aguilar-Ruiz, "A similarity-based approach for data stream classification," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4224–4234, 2014, doi: 10.1016/j.eswa.2013.12.041.
- [57] S. Huang and Y. Dong, "On mining time-changing data streams," *Chinese J. Electron.*, vol. 15, no. 2, pp. 220–224, 2006.
- [58] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "The CART decision tree for mining data streams," *Inf. Sci. (Ny)*, vol. 266, pp. 1–15, 2014, doi: 10.1016/j.ins.2013.12.060.
- [59] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 993–1004, 2018, doi: 10.1016/j.ejor.2017.08.040.
- [60] A. Bifet and R. Gavaldà, "Adaptive learning from evolving data streams," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5772 LCNS, pp. 249–260, 2009, doi: 10.1007/978-3-642-03915-7_22.
- [61] S. Xu and J. Wang, "A fast incremental extreme learning machine algorithm for data streams classification," *Expert Syst. Appl.*, vol. 65, pp. 332–344, 2016, doi: 10.1016/j.eswa.2016.08.052.
- [62] J. Gama, "A survey on learning from data streams: Current and future trends," *Prog. Artif. Intell.*, vol. 1, no. 1, pp. 45–55, 2012, doi: 10.1007/s13748-011-0002-6.
- [63] K. Prasanna Lakshmi and C. R. K. Reddy, "A survey on different trends in Data Streams," *ICNIT 2010 - 2010 Int. Conf. Netw. Inf. Technol.*, pp. 451–455, 2010, doi: 10.1109/ICNIT.2010.5508473.

- [64] V. S. Reddy, T. V Rao, and A. Govardhan, "Data mining techniques for data streams mining," *Rev. Comput. Eng. Stud.*, vol. 4, no. 1, pp. 31–35, 2017, doi: 10.18280/rces.040106.
- [65] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "NNs Architectures review," *Elsevier*, pp. 1–31, 2017, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231216315533>.
- [66] L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam-Moisy, "An introduction to deep learning," *ESANN 2011 - 19th Eur. Symp. Artif. Neural Networks*, pp. 477–488, 2011, doi: 10.1201/9780429096280-14.
- [67] S. H. Kim, C. Lee, and C. H. Youn, "An accelerated edge cloud system for energy data stream processing based on adaptive incremental deep learning scheme," *IEEE Access*, vol. 8, pp. 195341–195358, 2020, doi: 10.1109/ACCESS.2020.3033771.
- [68] A. V. Luong, T. T. Nguyen, and A. W.-C. Liew, "Streaming Active Deep Forest for Evolving Data Stream Classification," 2020, [Online]. Available: <http://arxiv.org/abs/2002.11816>.
- [69] N. Elsayed, A. S. Maida, and M. Bayoumi, "Sensor Sequential Data-Stream Classification Using Deep Gated Hybrid Architecture," *IEEE Green Technol. Conf.*, vol. 2019-April, pp. 1–4, 2019, doi: 10.1109/GreenTech.2019.8767136.
- [70] M. Hmayda, R. Ejbali, and M. Zaied, "Program classification in a stream TV using deep learning," *Parallel Distrib. Comput. Appl. Technol. PDCAT Proc.*, vol. 2017-Decem, pp. 123–126, 2018, doi: 10.1109/PDCAT.2017.00029.
- [71] P. Lara-Benítez, M. Carranza-García, J. García-Gutiérrez, and J. C.

- Riquelme, “Asynchronous dual-pipeline deep learning framework for online data stream classification,” *Integr. Comput. Aided. Eng.*, vol. 27, no. 2, pp. 101–119, 2020, doi: 10.3233/ICA-200617.
- [72] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, “A Hybrid Deep Learning based Model for Anomaly Detection in Cloud Datacentre Networks,” *IEEE Trans. Netw. Serv. Manag.*, vol. PP, no. c, p. 1, 2019, doi: 10.1109/TNSM.2019.2927886.
- [73] S. Dhote, C. Vichoray, R. Pais, S. Baskar, and P. Mohamed Shakeel, “Hybrid geometric sampling and AdaBoost based deep learning approach for data imbalance in E-commerce,” *Electron. Commer. Res.*, vol. 20, no. 2, pp. 259–274, 2020, doi: 10.1007/s10660-019-09383-2.
- [74] S. Young, T. Abdou, and A. Bener, *Deep super learner: A deep ensemble for classification problems*, vol. 10832 LNAI. Springer International Publishing, 2018.
- [75] Z. Chen *et al.*, “Feature selection may improve deep neural networks for the bioinformatics problems,” *Bioinformatics*, vol. 36, no. 5, pp. 1542–1552, 2020, doi: 10.1093/bioinformatics/btz763.
- [76] G. Sahebi, P. Movahedi, M. Ebrahimi, T. Pahikkala, J. Plosila, and H. Tenhunen, “GeFeS: A generalized wrapper feature selection approach for optimizing classification performance,” *Comput. Biol. Med.*, vol. 125, no. April, p. 103974, 2020, doi: 10.1016/j.compbiomed.2020.103974.
- [77] M. Bahri, “Improving IoT data stream analytics using summarization techniques Maroua Bahri To cite this version : HAL Id : tel-02865982 Improving IoT Data Stream Analytics Using Summarization Techniques,” 2020.

- [78] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, “Ensemble learning for data stream analysis: A survey,” *Inf. Fusion*, vol. 37, pp. 132–156, 2017, doi: 10.1016/j.inffus.2017.02.004.
- [79] M. A. Thalor and S. Patil, “Incremental learning on non-stationary data stream using ensemble approach,” *Int. J. Electr. Comput. Eng.*, vol. 6, no. 4, pp. 1811–1817, 2016, doi: 10.11591/ijece.v6i4.10255.
- [80] H. Yu, X. Sun, and J. Wang, “Ensemble OS-ELM based on combination weight for data stream classification,” *Applied Intelligence*, vol. 49, no. 6, pp. 2382–2390, 2019, doi: 10.1007/s10489-018-01403-2.
- [81] S. Wang, L. L. Minku, and X. Yao, “Resampling-based ensemble methods for online class imbalance learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, 2015, doi: 10.1109/TKDE.2014.2345380.
- [82] L. P. Cavaleiro, A. De Souza Britto, J. P. Barddal, and L. Heutte, “Dynamically Selected Ensemble for Data Stream Classification,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 2021-July. 2021, doi: 10.1109/IJCNN52387.2021.9533702.
- [83] J. K. B and M. Wo, “Multi Sampling Random Subspace Ensemble for Imbalanced,” pp. 360–369, 2020, doi: 10.1007/978-3-030-19738-4.
- [84] H. M. Gomes and F. Enembreck, “SAE: Social Adaptive Ensemble classifier for data streams,” *Proc. 2013 IEEE Symp. Comput. Intell. Data Mining, CIDM 2013 - 2013 IEEE Symp. Ser. Comput. Intell. SSCI 2013*, pp. 199–206, 2013, doi: 10.1109/CIDM.2013.6597237.
- [85] “Rule_based_design.pdf.” .
- [86] Q. Shen, R. Diao, and P. Su, “Feature Selection Ensemble,” vol. 10, pp. 289–

270, 2018, doi: 10.29007/rlxq.

- [87] J.-A. Romero-Navarrete and F. Otremba, *A computational scheme for assessing driving*, vol. 2. Springer International Publishing, 2019.
- [88] N. De Jay, S. Papillon-Cavanagh, C. Olsen, N. El-Hachem, G. Bontempi, and B. Haibe-Kains, “MRMRe: An R package for parallelized mRMR ensemble feature selection,” *Bioinformatics*, vol. 29, no. 18, pp. 2365–2368, 2013, doi: 10.1093/bioinformatics/btt383.
- [89] P. Yang, W. Liu, B. B. Zhou, S. Chawla, and A. Y. Zomaya, “Ensemble-based wrapper methods for feature selection and class imbalance learning,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7818 LNAI, no. PART 1, pp. 544–555, 2013, doi: 10.1007/978-3-642-37453-1_45.
- [90] H. Abubaker, A. Ali, S. M. Shamsuddin, and S. Hassan, “Exploring permissions in android applications using ensemble-based extra tree feature selection,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 1, pp. 543–552, 2020, doi: 10.11591/ijeecs.v19.i1.pp543-552.
- [91] A. K. Verma, S. Pal, and B. B. Tiwari, “Skin disease prediction using ensemble methods and a new hybrid feature selection technique,” *Iran J. Comput. Sci.*, vol. 3, no. 4, pp. 207–216, 2020, doi: 10.1007/s42044-020-00058-y.
- [92] S. Singh and S. Silakari, “An ensemble approach for feature selection of Cyber Attack Dataset,” *J. Comput. Sci.*, vol. 6, no. 2, p. 6, 2009.
- [93] G. Casalino, G. Castellano, and C. Mencar, “Data Stream Classification by Dynamic Incremental Semi-Supervised Fuzzy Clustering,” *Int. J. Artif. Intell. Tools*, vol. 28, no. 8, pp. 1–26, 2019, doi:

10.1142/S0218213019600091.

- [94] X. L. Li, P. S. Yu, B. Liu, and S. K. Ng, “Positive unlabeled learning for data stream classification,” *Soc. Ind. Appl. Math. - 9th SIAM Int. Conf. Data Min. 2009, Proc. Appl. Math.*, vol. 1, pp. 256–267, 2009, doi: 10.1137/1.9781611972795.23.
- [95] C. SAC '13. (2013 and Association for Computing Machinery., “Proceedings of the 28th Annual ACM Symposium on Applied Computing.,” pp. 801–806, 2013.
- [96] Z. Liu *et al.*, “Self-paced ensemble for highly imbalanced massive data classification,” *Proc. - Int. Conf. Data Eng.*, vol. 2020-April, pp. 841–852, 2020, doi: 10.1109/ICDE48307.2020.00078.
- [97] B. Junior and M. do C. Nicoletti, “An iterative boosting-based ensemble for streaming data classification,” *Inf. Fusion*, vol. 45, no. December 2017, pp. 66–78, 2019, doi: 10.1016/j.inffus.2018.01.003.
- [98] K. K. Wankhade, K. C. Jondhale, and S. S. Dongre, “A clustering and ensemble based classifier for data stream classification,” *Appl. Soft Comput.*, vol. 102, p. 107076, 2021, doi: 10.1016/j.asoc.2020.107076.
- [99] M. Arya and H. Sastry G, “A Novel Deep Ensemble Learning Framework for Classifying Imbalanced Data Stream,” in *IOT with Smart Systems*, Springer, Singapore, 2022, pp. 607–617.
- [100] M. Arya and H. Sastry G, “DEAL–‘Deep Ensemble ALgorithm’ Framework for Credit Card Fraud Detection in Real-Time Data Stream with Google TensorFlow,” *Smart Sci.*, vol. 8, no. 2, pp. 71–83, 2020, doi: 10.1080/23080477.2020.1783491.

- [101] “NSE website.” <https://www1.nseindia.com/>.
- [102] “Finance yahoo website.” .
- [103] M. Arya and H. Sastry G, “Stock Indices Price Prediction in Real Time Data Stream using Deep Learning with Extra-Tree Ensemble (DELETE) Optimization.” doi: 10.1504/IJCSE.2021.10040278.
- [104] A. F. Agarap, “An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification,” pp. 5–8, 2017, [Online]. Available: <http://arxiv.org/abs/1712.03541>.
- [105] U. Seiffert, “Multiple layer perceptron training using genetic algorithms,” *Eur. Symp. Artif. Neural Networks*, no. April, pp. 159–164, 2001, [Online]. Available:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.6859&rep=rep1&type=pdf>.
- [106] S. M. Taheri and G. Hesamian, “A generalization of the Wilcoxon signed-rank test and its applications,” *Stat. Pap.*, vol. 54, no. 2, pp. 457–470, 2013, doi: 10.1007/s00362-012-0443-4.

An ensemble-based algorithm for Efficient classification of real time data streams

ORIGINALITY REPORT

9%

SIMILARITY INDEX

PRIMARY SOURCES

1	upcommons.upc.edu Internet	733 words — 3%
2	hwbdocuments.env.nm.gov Internet	187 words — 1%
3	nasri-narc.gov.np Internet	184 words — 1%
4	www.coursehero.com Internet	129 words — 1%
5	www.ctcswp.ca Internet	78 words — < 1%
6	www.ncbi.nlm.nih.gov Internet	46 words — < 1%
7	www.geeksforgeeks.org Internet	41 words — < 1%
8	export.arxiv.org Internet	34 words — < 1%
9	www.tandfonline.com Internet	34 words — < 1%

10	arxiv.org Internet	32 words — < 1%
11	"Progress in Computing, Analytics and Networking", Springer Science and Business Media LLC, 2020 Crossref	30 words — < 1%
12	Saliha Özarslan, Serdar Abut, M.R. Atelge, M. Kaya, S. Unalan. "Modeling and simulation of co-digestion performance with artificial neural network for prediction of methane production from tea factory waste with co-substrate of spent tea waste", Fuel, 2021 Crossref	27 words — < 1%
13	www.ijrte.org Internet	26 words — < 1%
14	medium.com Internet	25 words — < 1%
15	tel.archives-ouvertes.fr Internet	24 words — < 1%
16	www.hindawi.com Internet	23 words — < 1%
17	Mintu Pal, Thingreila Muinao, Hari Prasanna Deka Boruah, Neeraj Mahindroo. "Current advances in prognostic and diagnostic biomarkers for solid cancers: Detection techniques and future challenges", Biomedicine & Pharmacotherapy, 2022 Crossref	21 words — < 1%
18	Salah Ud Din, Junming Shao, Jay Kumar, Cobbinah Bernard Mawuli, S. M. Hasan Mahmud, Wei Zhang, Qinli Yang. "Data stream classification with novel class	20 words — < 1%

detection: a review, comparison and challenges", Knowledge and Information Systems, 2021

Crossref

19 Julián Darío Miranda-Calle, Vikranth Reddy C., Parag Dhawan, Prathamesh Churi. "Exploratory data analysis for cybersecurity", World Journal of Engineering, 2021

19 words — < 1%

Crossref

20 Shuliang Xu, Junhong Wang. "Dynamic extreme learning machine for data stream classification", Neurocomputing, 2017

19 words — < 1%

Crossref

21 Tanvi Patel, Devkishan Patel, Taral Patel. "Chapter 56 Breast Cancer Prediction Analysis Using Data Mining Techniques", Springer Science and Business Media LLC, 2022

19 words — < 1%

Crossref

22 baadalsg.inflibnet.ac.in

Internet

19 words — < 1%

23 semspub.epa.gov

Internet

19 words — < 1%

24 Ehsan Eslami, Mahdi Eftekhari. "An effective hybrid model based on PSO-SVM algorithm with a new local search for feature selection", 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), 2014

18 words — < 1%

Crossref

25 Karthikeyan, S., P. Asokan, S. Nickolas, and Tom Page. "Solving flexible job-shop scheduling problem using hybrid particle swarm optimisation algorithm"

16 words — < 1%