3-2019

Streaming Feature Grouping and Selection (Sfgs) For Big Data Classification

Noura Helal Hamad Al Nuaimi

UAEU

جامعة الإمارات العربيـة المتحدة
United Arab Emirates University

United Arab Emirates University

College of Information Technology

STREAMING FEATURE GROUPING AND SELECTION (SFGS) FOR BIG DATA CLASSIFICATION

Noura Helal Hamad Khudouma Al Nuaimi

This dissertation is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Under the Supervision of Dr. Mohammad Mehedy Masud

March 2019

Declaration of Original Work

I, Noura Helal Hamad Khudouma Al Nuaimi, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this dissertation entitled "*Streaming Feature Grouping and Selection (SFGS) for Big Data Classification*", hereby, solemnly declare that this dissertation is my own original research work that has been done and prepared by me under the supervision of Dr. Mohammad Mehedy Masud, in the College of Information Technology at UAEU. This work has not previously been presented or published, or formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my dissertation have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this dissertation.

Student's Signature: نوره النعيمي Date: 5/5/2019

Approval of the Doctorate Dissertation

This Doctorate Dissertation is approved by the following Examining Committee Members:

1) Advisor (Committee Chair): Mohammad Mehedy Masud
   Title: Associate Professor
   Department of Information Systems and Security
   College of Information Technology

   Signature ______________________ Date 28-03-2019

2) Member: Amir Ahmad
   Title: Assistant Professor
   Department of Information Systems and Security
   College of Information Technology

   Signature ______________________ Date 28-03-2019

3) Member: Mamoun Awad
   Title: Associate Professor
   Department of Computer Science and Software Engineering
   College of Information Technology

   Signature ______________________ Date 28/3/2019

4) Member (External Examiner): Nizar Bouguila
   Title: Professor
   Department of Concordia Institute for Information Systems Engineering
   Institution: Concordia University

   Signature ______________________ Date 28-03-2019

This Doctorate Dissertation is accepted by:

Acting Dean of the College of Information Technology: Professor Khaled Shuaib

Signature ______________________ Date 5-5-2019

Acting Dean of the College of Graduate Studies: Professor Ali Al Marzouqi

Signature Ali Hassan Date 5/5/2019

Copy 7 of 7

Advisory Committee

1) Advisor: Mohammad Mehedy Masud

Title: Associate Professor

Department of Information Systems and Security

College of Information Technology

2) Co-advisor: Mohamed Adel Serhani

Title: Associate Professor

Department of Information Systems and Security

College of Information Technology

3) Member: Nazar Zaki

Title: Professor

Department of Computer Science and Software Engineering

College of Information Technology

Abstract

Real-time data has always been an essential element for organizations when the quickness of data delivery is critical to their businesses. Today, organizations understand the importance of real-time data analysis to maintain benefits from their generated data. Real-time data analysis is also known as real-time analytics, streaming analytics, real-time streaming analytics, and event processing. Stream processing is the key to getting results in real time. It allows us to process the data stream in real time as it arrives. The concept of streaming data means the data are generated dynamically, and the full stream is unknown or even infinite. This data becomes massive and diverse and forms what is known as a big data challenge. In machine learning, streaming feature selection has always been a preferable method in the preprocessing of streaming data. Recently, feature grouping, which can measure the hidden information between selected features, has begun gaining attention. This dissertation's main contribution is in solving the issue of the extremely high dimensionality of streaming big data by delivering a streaming feature grouping and selection algorithm. Also, the literature review presents a comprehensive review of the current streaming feature selection approaches and highlights the state-of-the-art algorithms trending in this area. The proposed algorithm is designed with the idea of grouping together similar features to reduce redundancy and handle the stream of features in an online fashion. This algorithm has been implemented and evaluated using benchmark datasets against state-of-the-art streaming feature selection algorithms and feature grouping techniques. The results showed better performance regarding prediction accuracy than with state-of-the-art algorithms.

Keywords: Stream of features, features grouping, feature selection, relevance analysis, redundancy analysis.

Title and Abstract (in Arabic)

ميزة تجميع التدفق والاختيار (SFGS) لتصنيف البيانات الضخمة

الملخص

استطاع مفهوم تحليل البيانات اليوم أن يفرض نفسه على كثير من التخصصات المختلفة ذات المجالات المتنوعة، حيث أصبحت المؤسسات تدرك أهمية تحليل البيانات فورياً في تطوير خدماتها أو منتجاتها أوما يتعلق بأي منهما. من هنا ظهر علم (streaming feature selection) والذي يعتبر أحد التخصصات المدرجة في مجال معالجة البيانات المتدفقة، حيث تعتبر بيانات متدفقة بصورة هائلة يصعب التنبؤ بحجمها أو حتى حصرها. تشكل عملية فرز البيانات الخطوة الأولى في اختيار المفيد منها بطريقة علمية مقنّنة، وذلك لتحقيق هدف اكتشاف الحقائق الخفية في قواعد البيانات. في الآونة الأخيرة برزت كفاءة وأهمية (feature grouping) في تعزيز قدره (feature selection) الانتقائية، حيث تعتمد فكرته على تجميع (features) إلى مجموعات أصغر وانتقاء الأكثر فائدة من بينها. في هذا البحث سوف نتناول مشكلة البيانات الضخمة المتولدة والمتدفقة بشكل مستمر، بالإضافة إلى كيفية فرز هذه البيانات فورياً، وذلك للمساهمة في دعم اتخاذ القرارات التنفيذية المستقبلية. يقدم البحث حلاً لعملية فرز هذه البيانات، بحيث يعتمد على مفهوم التجميع للميزات المتدفقة. بداية تعرض الدراسة الأدبية في بحثنا هذا مسحاً دقيقاً لنظريات ونماذج تم جمع بياناتها من الدوريات والنشرات الرسمية وبعض المصادر العلمية الأخرى، بالإضافة إلى بعض الخوارزميات المعمول بها في مجال (streaming feature selection). كما ويستعرض البحث دراسة تجريبية مفصلة لغرض المقارنة وإثبات كفاءة الطريقة المقترحة عملياً وذلك باستخدام بيانات مرجعية ومقارنتها بخوارزميات أخرى. حيث أظهرت النتائج أداءً فائقًا فيما يتعلق بدقة التنبؤ.

مفاهيم البحث الرئيسية: تدفق الميزات، تجميع الميزات ، انتقاء الميزات ، تحليل العلاقة ، تحليل التكرار.

Acknowledgements

I gratefully thank ALLAH, the Might, for giving me continual motivation, patience, and abilities to complete my dream study while earning my PhD degree.

Sincere thanks are due to my principal advisor, Dr. Mohammad Mehedy Masud, for his patience, motivation, and sharing his immense knowledge in data mining research, which enriched my dissertation outcomes. I am also grateful to my advisory committee members, Prof. Nazar Zaki and Dr. Mohamed Serhani, for their patience and continuous support in overcoming numerous obstacles that I encountered across the research phases.

I am also grateful to Ms. Maryam Al Mandhri for her unfailing support and assistance, along with the DVCRGS for its fellowship grant for the PhD degree study. My sincere thanks also goes to the library, which supported my work and helped me finalize a higher-quality thesis. Special appreciation is due to Mr. Ahmed Taha (Library Research Desk) for his unique dissertation services. I would also like to thank many others who genuinely shared their knowledge with me during the last six years.

Finally, I must express my very profound gratitude to my parents, brothers, and sisters for providing me unfailing support and continuous encouragement. This accomplishment would not have been possible without your support.

Thank you all

Dedication

To my beloved parents and family

Table of Contents

List of Tables

List of Figures

List of Abbreviations

C4.5	Is an extension of Quinlan's earlier ID3 algorithm which is used to generate a decision tree
CFS	Correlation-based Feature Selection
CMIM	Conditional Mutual Information Maximization
CPU	Central Processing Unit
CVNS	Feature selection algorithm
DIA-RED	Dimension Incremental Algorithm for Reduction Computation
EVMS	Eastern Virginia Medical School
FAST	Fast Clustering-based Feature Selection Algorithm
FAST	Feature Subset Selection Algorithm
FCBF	Fast Correlation Based Filter
GFLasso	Graph-guided Fused Lasso
GR	Gain Ratio
Hybrid FCBF	A hybrid Feature Selection (FS) system which combines Fast Correlation-Based Filter
ICU	Intensive Care Unit
ICUs	Intensive Care Units
ID3	Is an algorithm used to generate a decision tree from a dataset
IG	Information Gain
J48	An open source Java implementation of the C4.5 decision tree algorithm
K-NN	K-Nearest Neighbor
LFI	Learning Features Added Individually
LGF	Learning Grouped Features Added Sequentially
LOFS	Open-source Library for Feature Selection

M	Dimensionality
Maximization(CMIM)	Maximization Conditional Mutual Information
MIFS	Mutual Information Feature Selection
MIMIC-II	Clinical Database contains comprehensive clinical data from the Intensive Care Unit (ICU) patients
mRMR	Minimum Redundancy Maximum Relevance
N	Size of Sample
NCI	National Cancer Institute
NIPS	Neural Information Processing Systems Conference
OGFS	Online Group Feature Selection
OSFS	Online Streaming Feature Selection
OS-NRRSAR-SA	Extension for online streaming feature selection (OSFS) from the rough sets (RS) perspective
PCA	Principal Component Analysis
PGVNS	Predominant Group Based Variable Neighborhood
RS	Rough Sets
SAOLA	Scalable and Accurate Online Approach
SFFS	Sequential Forward Floating Search
SFGS	Streaming Feature Grouping and Selection
SMO	Sequential Minimal Optimization
SU	Symmetrical Uncertainty
SVM	Support Vector Machine
WEKA	Waikato Environment for Knowledge Analysis is a suite of machine learning software
UCI	The UCI Machine Learning Repository for public databases

Chapter 1: Introduction

1.1 Overview

Today, many organizations in various domains are continuously generating heterogeneous data in real time in considerable volume. This big data is naturally arriving as a never-ending stream of events. Therefore, the world is predicted to create about 180 zettabytes of data (or 180 trillion gigabytes) in 2025 [1]. These streaming big data could be analyzed in real time to attain a strategic value and build competitive advantages. Among these advantages would be supporting, efficient decision-making processes, with which most organizations hope to gain a significant advantage. Currently, streaming data is challenging traditional machine learning to present more suitable approaches, especially with the challenge of big data.

Feature selection techniques are an important part of machine learning. Feature selection is also known as variable selection, attribute selection, or variable subset selection. Moreover, it is the first option to reduce the extreme size of streaming data. It is used to reduce the input features and find the most informative ones to be used in model construction. Feature selection techniques should be distinguished from feature extraction even though both techniques are used to reduce the dimensionality. Feature extraction transforms raw data into features suitable for modeling, but feature selection removes unnecessary features. For example, principal component analysis (PCA) combines similar (correlated) attributes and creates new ones. Feature extraction is a dimensionality reduction technique that creates new combinations of attributes whereas feature selection methods include and exclude attributes present in the data without changing them.

In streaming data, there are two categories: instances stream and feature stream. In instances stream, it is assumed that instances arrive continuously, one after another, over time, and the number of features is fixed. In contrast, in the stream of features, it is assumed that features arrive continuously, one after another, over time, but the number of instances is fixed. Feature selection with streaming features is known as streaming feature selection, or in other literature, online streaming feature selection. It is a popular approach for reducing streaming data size by selecting the most informative features. Streaming feature selection practices introduce more efficient methodologies that can handle the rapid growth of data volumes over time.

1.2 Motivation

In streaming feature selection, candidate features arrive sequentially; however, the size of features is unknown. Streaming feature selection has recently gotten attention in the field of real-time application. Streaming feature selection has a critical role in real-time applications, in which the required action must be taken or a decision made very quickly or within a specific timeframe. In applications such as weather forecasting, transportation systems, stock markets, clinical research in real-time, natural disasters, call records, and vital-sign monitoring, streaming feature selection plays a crucial role in preparing data for the analysis process efficiently and effectively. Furthermore, the challenge of analyzing and mining terabytes and petabytes of data is a tedious task with traditional data mining techniques, along with the challenge of real-time analytics. For example, it might want to analyze a user's political behavior on Twitter, on which the user produces many tweets per day, including new words and abbreviations (i.e., unlimited features). Using a regular batch feature selection would be challenging with the sequence of tweets. Another example is bioinformatic and clinical machine learning problems, where acquiring the entire set of features for every training instance is expensive due to the high cost of laboratory experiments [2].

The last example is the wind-power problem in weather forecasting [3][4], which is today considered one of the most important sources of renewable energy. The researchers may deploy a set of observation stations in specific areas. Each station is treated as an instance, and the total number of stations represents the total number of instances. In contrast, the number of features continues increasing with each new observation over time. Even though the data collection rate may not be very high, the underlying dataset's dimensionality can easily reach tens or hundreds of thousands after a while. The challenge is determining how to predict the generated power in wind farms.

1.3 Problem Statement

Several methods of streaming feature selection have been proposed to address the streaming feature selection problem. However, it is understood that it is still facing a shortage of efficient techniques that could handle the extremely high dimensionality of streaming data. The problem this research attempts to address is *how to resolve the feature selection problem with streaming features in the context of streaming big data.*

Let the *stream of features* be $\{ f_1, f_2, \dots , f_i\}$. How can the most informative features be selected? $G_i = \{f_{i_1}, \dots, f_{i_n}\}, where\ f_{i_j}$is j^{th} feature of group G_i? Accordingly, how this group be updated, G_i', when it receives a new feature, f_{i+1}?

1.4 Aim of Research

This dissertation introduces a novel and efficient feature selection technique to reduce the extremely high dimensionality of streaming big data. The study achieved the research aim by reaching specific objectives:

1. Identifying the irrelevant and redundant features in the features stream.
2. Selecting the most informative features in the features stream and building a learning model.
3. Introducing efficient analysis of feature relevance and redundancy to handle the high dimensionality of streaming big data.

1.5 Research Questions

In this research, a propose solution for the problem of the high dimensionality of streaming big data for a classification problem is presented. The following research questions are raised to address this problem and achieve the dissertation’s objectives (Figure 1):

1. How could irrelevant features be removed from the features stream?
2. How could redundant features be removed from the features stream?
3. How could the streaming feature selection efficiency be achieved?

For the first two questions, it need to be distinguish between the relevant feature concept from the redundant feature concept. Each represents a specific criterion to evaluate the feature at the stream of a feature. A relevant feature means that the new, coming feature is relevant to the class because it is a classification problem. A redundant feature means that the new feature is not redundant to any existing features.

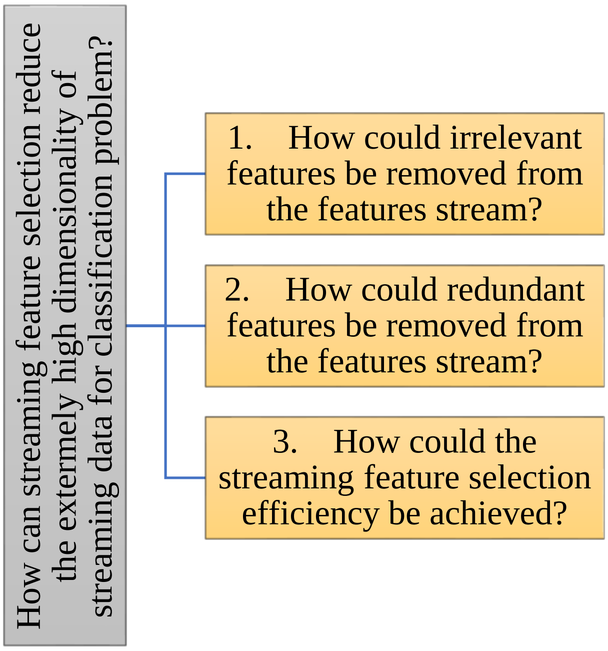
How can streaming feature selection reduce the extermely high dimensionality of streaming data for classification problem?
1. How could irrelevant features be removed from the features stream?
2. How could redundant features be removed from the features stream?
3. How could the streaming feature selection efficiency be achieved?

Figure 1: Research’s problem and the three proposed research’s questions

1.6 Dissertation Contribution

The core contribution of this dissertation is its delivery of a new methodology to handle streaming feature selection called streaming feature grouping and selection (SFGS). Two elements distinguish feature selection in streaming features from traditional feature selection. The first distinction is that the total number of features is unknown over time and could be infinite. In other words, no total number of features exists. Second, any new feature from the feature stream requires online inspection upon its arrival. This dissertation started with a comprehensive review of the current approaches and highlighted the state-of-the-art algorithms trending in this area. Detailed design and algorithm characteristics have been shown to promote the algorithm approach.

The dissertation’s second contribution is its confrontation of the challenge of reducing the extremely high dimensionality of streaming data for classification. The significant characteristic of the proposed method is that it can handle the extremely high dimensionality of streaming data with various types and sizes of datasets. Also the dissertation investigates the applicability of the proposed approach to be used in the future in the case of big data.

The third contribution is delivering the SFGS that could be integrated with real-world applications that manipulate real-time data. Thus, it could support these applications in conducting real-time analytics more efficiently.

Finally, the SFGS is evaluated with real data and compared to state-of-the-art algorithms. In the experiment, SFGS was evaluated using public challenge datasets as a benchmark and compared with three state-of-the-art algorithms: predominant group-based variable neighborhood search (PGVNS), Fast-OSFS and Alpha investing. The resulting prediction accuracy and running time were mostly better than, or at least as good as, other algorithms.

1.7 Dissertation Structure

The dissertation is structured based on the research questions. Accordingly, the dissertation comprises the following chapters:

Chapter 2: Literature Review

In the second chapter, a review of the existing literature related to traditional feature selection algorithms is provided. Also, a study of the current algorithms that use streaming feature selection to determine their strengths and weaknesses is presented too. Furthermore, the chapter demonstrates the related definitions and sheds light on the ongoing challenges in big data research.

Chapter 3: Traditional Feature Selection and Predicting Patient Deterioration: Study Case

In this chapter, potential research areas in the feature selection scope is explored. It examines the traditional feature selection role in reducing the high dimensionality of streaming data. Furthermore, it illustrates the implementation of feature selection for a specific scenario, which is predicting ICU patient deterioration.

Chapter 4: Proposed Streaming Feature Grouping and Selection Approach

The proposed technique, called the streaming-feature grouping and selection (SFGS) approach, is described in this chapter. Also, the chapter presents the SFGS algorithm pseudocode and illustrates two scenarios for running the method. Besides, it analyzes the proposed approach’s q-factor and discusses the runtime complexity.

Chapter 5: Experimentation and Evaluation

In this chapter, the SFGS experimental work is demonstrated. It starts by presenting detailed information about the datasets, learning algorithms, the three competing state-of-the-art approaches, the hardware and software environments and the setup of parameters. It also reports and evaluates the experiment's results. It compares the proposed algorithm's prediction accuracy with that of the three competing algorithms. In addition, It analyzes execution time to evaluate the proposed algorithm's performance against the competing approaches. Last, it examines the sensitivity of the parameters that could affect the proposed approach's prediction accuracy.

Chapter 6: Conclusion and Future Research

This chapter ends with the conclusion of this dissertation. Additionally, it provides the recommendations stemming from this research. It furnishes some recommendations to improve and extend the SFGS approach. It also suggests future research.

In the next chapter, the existing relevant literature is reviewed to highlight the research gap, support the thesis research issues and justify the generated findings.

Chapter 2: Literature Review

In this chapter, the literature is reviewed. It compares studies in the context of the classification problem. It starts by reviewing the traditional feature selection algorithms and then explores the strengths and weaknesses of the current algorithms of streaming feature selection. The chapter also sheds light on the ongoing challenges in big data research.

Feature selection techniques are an essential part of machine learning. Feature selection is often termed as variable selection, attribute selection and variable subset selection. It is the process of reducing input features to the most informative ones for use in model construction. Feature selection should be distinguished from feature extraction. Although, both techniques are used to reduce the number of features in a dataset, feature extraction is reduction technique in dimensionality that creates new combinations of attributes, whereas feature selection includes and excludes the attributes that are present in the data without changing them.

Streaming feature selection has recently received attention concerning real-time applications. Feature selection with streaming data, known as streaming feature selection or online streaming feature selection is a preferred technique that uses a selection of features that are most informative to reduce streaming data size.

In streaming feature selection, the candidate features arrive sequentially. The size of these features is unknown. Streaming feature selection has a critical role in real time applications, for which the required action must be taken immediately. In applications such as weather forecasting, transportation, stock markets, clinical research, natural disasters, call records, and vital-sign monitoring, streaming feature selection plays a crucial role in efficiently and effectively preparing data for the analysis process in real time.

At present, contemporary methods in machine learning are being challenged by streaming data as newer and faster algorithms deal with variable volumes of data. Making decisions in real time from such continuous data could bring data monetization benefit which is a significant source of revenue. The world is projected to generate over 180 zettabytes (or 180 trillion gigabytes) of data by 2025 [1]. This figure when compared with 10 zettabytes worth data created as of 2015 seems ubiquitous. The presence of large datasets is the reason for the emergence artificial intelligence. Companies such as Google, Facebook, Baidu, Amazon, IBM, Intel, and Microsoft are investing in capturing talent pool to understand big data and release open artificial intelligence hardware and software [1].

Using big data for streaming feature selection is regarded as a solution to select the most informative features that could support the development of robust and accurate machine learning models. There are several techniques in data analytics. The newer algorithms on dimensionality reduction are asymptotically better than the previous algorithms. Prior research on feature selection has targeted searching for relevant features only. John et al. [5] proposed three categories belonging to X input features and its importance in C target class: (1) strongly relevant, (2) weakly relevant, and (3) irrelevant. Yu and Liu [6] improved this categorization by proposing a definition of feature redundancy therefore creating a path for efficient elimination of redundant features.

Let F be a full set of features, F_i a feature and $S_i = F - \{F_i\}$. The definitions and techniques are listed as follows:

Definition 1 (Strong relevance): Feature F_i is strongly relevant if and only if

$P\ (C \mid F_i\ S_i\ )\ \neq P\ (\ C\ |S_i\ )$. (1)

Thus, a feature with strong relevance will always be in the final, optimal feature subset.

Definition 2 (Weak relevance): Feature F_i is weakly relevant if and only if

$P\ (C \mid F_i, S_i\ )\ = P\ (\ C\ |S_i\ )$, and $\exists\ S_i' \subset S_i$, such that $P\ (C \mid F_i, S_i'\ )\ \neq P\ (\ C\ |S_i'\ )$. (2)

A feature with weak relevance is not always in the final, optimal feature subset, but ideally, it would be included.

Definition 3 (Irrelevance): Feature F_i is irrelevant if and only if

$\forall\ S_i' \subseteq S_i, P\ (C \mid F_i, S_i'\ )\ = P\ (\ C\ |S_i'\ )$. (3)

Irrelevant features are not necessary at all and thus should be discarded.

According to Yu and Liu [6] the important and relevant features are segregated into necessary and unnecessary features. Yu and Liu's definition [6], which is based on Markov blanket is that redundant features provide no extra information than the currently selected features and irrelevant features provide no useful information in the final model. The definition from other authors is given below:

Definition 4 (Markov blanket): Given a feature F_i, let $M_i \subset F\ (F_i\ \notin\ M_i\ ), M_i$ is said to be a Markov blanket for F_i if and only if

$P(F - M_i - \{F_i\}, C\ |F_i, M_i) = P\ (F - M_i - \{F_i\}, C \mid M_i)$ (4)

Definition 5 (Redundant feature): Let G be the current set of features. A feature is redundant and hence needs to be removed from G if and only if there is a weak relevance and has a Markov blanket M_i within G.

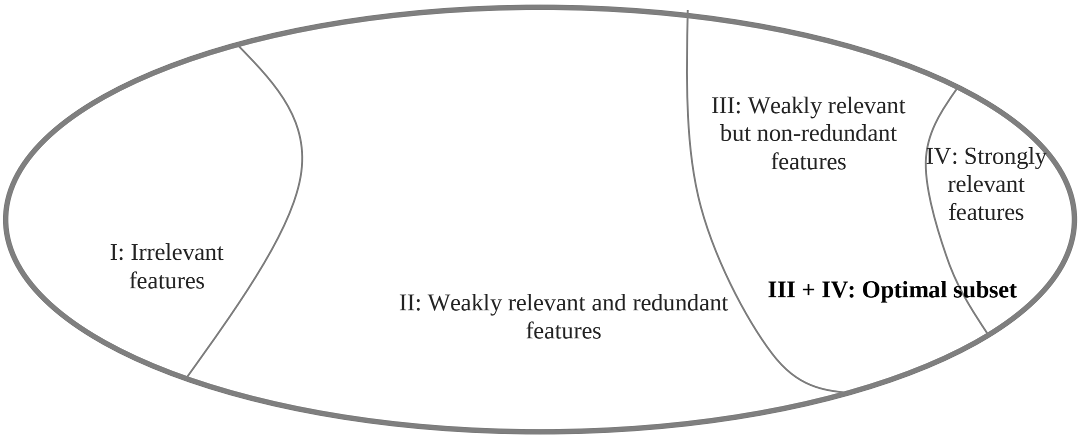
I: Irrelevant features
II: Weakly relevant and redundant features
III: Weakly relevant but non-redundant features
IV: Strongly relevant features
III + IV: Optimal subset

Figure 2: Feature relevance and redundancy relationships

Figure 2 shows the relationship between redundancy and the importance of a feature. The figure shows segregation of entire feature sets into four disjointed subsets comprising of a) irrelevant feature (I) b) redundant features (II) and less relevant features c) less relevant but non-redundant features (III) and d) features that are strongly relevant (IV). It also depicts an optimal subset having features of both (III) and (IV). It is necessary to mention that parts (II) and (III) are disjointed but multiple partitions of these parts can form due to Markov-blanket filtering.

In systems based on machine learning, streaming feature selection sometimes referred to as Online Streaming Feature Selection (OSFS) or online feature selection is a method used to choose a group of essential features (e.g. variable X or multiple predictors) from streaming data to construct a theoretical model. Streaming feature selection allows for the most informative features to be selected by eliminating redundant and irrelevant features. In comparison with older feature selection methods, online feature selection leads to (a) models that are easier for researchers and users to interpret (b) lesser training time, avoiding issues and challenges related to dimensionality and (c) greater generalization through reduced over-fitting [7]. Figure 3 illustrates the feature selection classification of data from two perspectives:

static feature selection and streaming feature selection. In static data, all features and instances of data are assumed to be captured well in advance, whereas streaming data has unknown numbers of data instances, features or both.

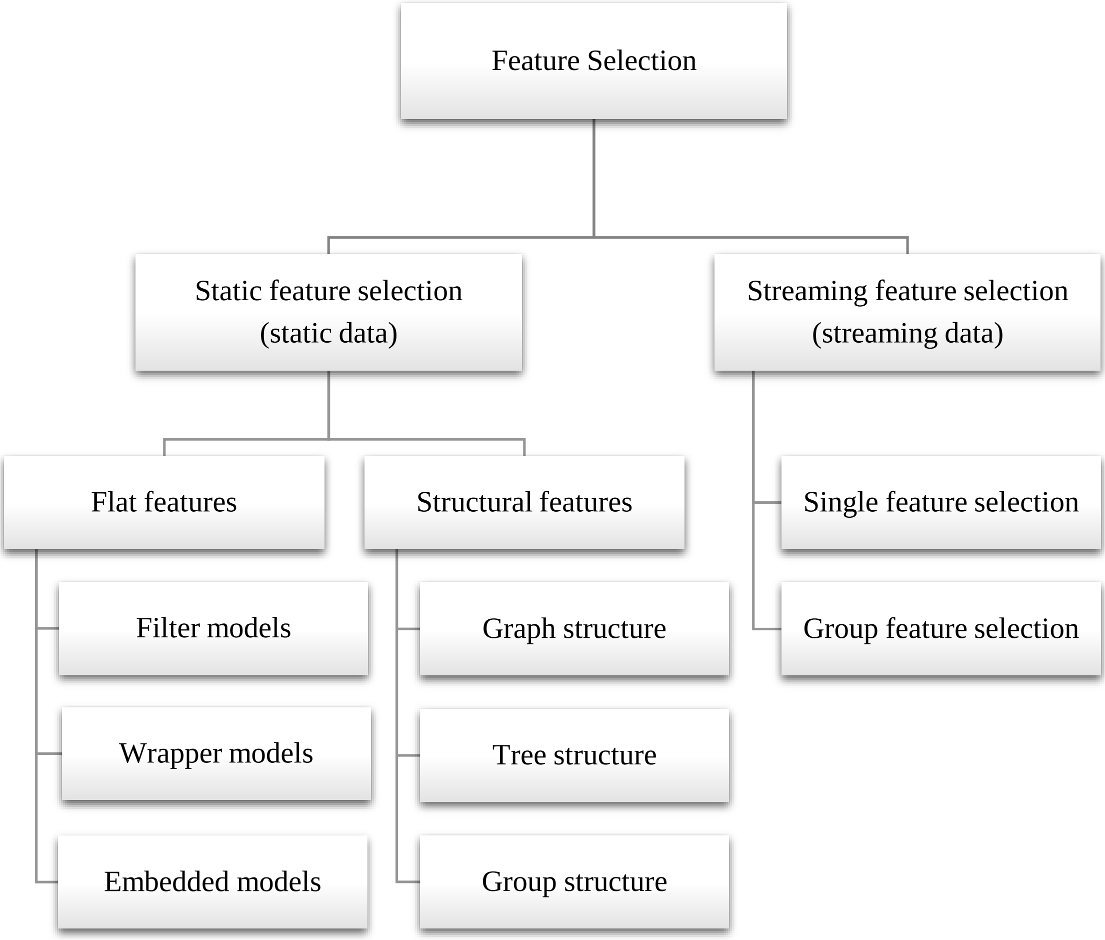

Figure 3: Feature-selection classification taxonomy

2.1 Static Feature Selection

From the features' perspective, static features can be categorized as flat features or structured features. Flat features are independent. However, structures features are usually in the form of the graph structure, tree structure or group structure. A conventional approach to feature-selection is aimed at working with flat features which can be regarded as independent. Algorithms in the flat-features category are subcategorized into three main groups: filters, wrappers, and embedded models.

2.1.1 Flat Features

Filter Methods

Feature selection focus on the application of statistical measures for assigning scores for each feature. This is followed by score based feature ranking that may be selected or removed from the datasets. The methods are sometimes univariate and could consider the features independently or about the dependent variable, as shown in Figure 4. Famous algorithms from this category include the Fisher score [8][9], information theory based methods[10]–[12], and ReliefF and its variants [13][14].

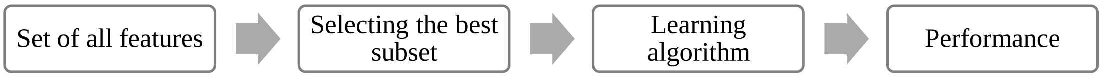

Figure 4: Filter method process

The Fisher score, also known as the scoring algorithm [8][9] is a form of Newton’s method used in statistics to numerically solve maximum likelihood equations. It is named after Ronald Fisher. Fisher score is widely used for supervised feature selection methods. It selects each feature independently according to their scores under the Fisher criterion, which leads to a suboptimal subset of features [15].

Information theory based methods which are represented as a family consisting of feature selection algorithms are primarily methods that have its antecedents in information theory as shown in Table 1. In probability and information theory, the amount of information that two random variables share is affected by their mutual dependence.

Table 1: Categories of information theory-based methods

Ref.	**Information method**	**Description**
[16]	Mutual information maximization (or information gain)	Mutual information maximization (also known as information gain) feature importance level by its correlation with a class label. The assumption of this method is that in the event of a feature having strong correlation with a class label, it can be used to accomplish good classification performance.
[17]	Mutual information feature selection (MIFS)	MIFS was introduced to resolve the limitation of mutual information maximization. It can take into consideration feature relevancy and feature redundancy at the same time during feature selection phase.
[11]	Minimum redundancy maximum relevance (mRMR)	To reduce the effect of feature redundancy, mRMR is used to select features that have a high correlation with the class (output) and low correlations among themselves.
[18]	Conditional infomax feature extraction	Conditional infomax feature extraction was introduced to resolve the gaps in both MIFS and mRMR, which both consider feature relevance and feature redundancy at the same time. This method assumes that given the class labels if feature redundancy is stronger than intra-feature redundancy then there is a negative effect on feature selection.
[19]	Joint mutual information	Since MIFS and mRMR are useful in lowering feature redundancy during the process of feature selection, this alternative method known as joint mutual information was recommended to increase the sharing of complementary information between a new unselected feature and the selected feature when the class labels are given.

Table 1: Categories of information theory-based methods (continued)

Ref.	**Information method**	**Description**
[20]	Conditional mutual information maximization (CMIM)	In CMIM, features are iteratively selected to enhance the sharing of mutual information with class labels when the selected features are given. In other words, CMIM does not select the feature that is most similar to the previously selected ones, even though the predictive power of that feature for the class labels would be strong.
[21]	Informative fragments	The intuition behind informative fragments is that adding a new feature should maximize the value of conditional information that the new feature and the existing features share rather than the information that the features and the class share.
[22]	Interaction capping	Interaction capping is similar to CMIM, but instead of restricting the formula, interaction capping is non-negative.
[23]	Double input symmetrical relevance	Another type of information theory based method known as double input symmetrical relevance takes advantage of normalization approaches to normalize mutually exclusive information.
[12]	Fast correlation based filtering (Yu and Liu, 2003)	This filtering method takes advantage of feature-feature and feature-class correlations at the same time, using feature selection methods that cannot be turned into a unified conditional likelihood maximization framework easily.

ReliefF and its variant feature-selection algorithms are used in the binary classification that Kira and Rendell proposed in 1992 [24], features having high quality should give matching values to instances belonging to the same class and non-matching values in case instances belong to different classes. The strengths of these methods that they are not reliant on heuristics, they run in low-order polynomial time, and they are noise-tolerant and robust to feature interactions. Besides that, they are applicable to both binary and continuous data. Conversely, ReliefF will not discriminate among the existing redundant features and it is easy to fool the algorithm by using less number of instances [13][14]. According to Kononenko [25], the reliability of the probability approximation of the ReliefF algorithm can be improved through some updates and made more resilient to incomplete data. Therefore, this problem is considered as multi-class problem.

In recent works, scholars have proposed feature grouping to pinpoint groups with correlated features. This is an innovative method as it reduces the multi-dimensionality of large datasets. I highlight some of these efforts below. Among one of the strategies that uses feature grouping for increasing the efficiency of the feature search is called a predominant group based variable neighborhood search (PGVNS) [26]. PGVNS uses approximate Markov blanket and a predominant feature. García-Torres et al. [26] also introduced the concept of predominant groups and argued in favor of a heuristic strategy called GreedyPGG that group input space. While conducting the experiment they used synthetic and real datasets obtained from microarray and text-mining domains. The results were compared with fast correlation based filter (FCBF) [6], the fast clustering-based feature-selection algorithm (FAST) [27], and CVNS [26] which are the three popular algorithms for feature selection.

Gangurde [28] and Gangurde and Metre [29] have argued in favor of a clustering concept that uses feature selection to handle the issue of dimensionality reduction in big data. A minimum spanning tree is used to create a cluster formation therefore reducing the computational complexity of feature selection. However, the study primarily deals with the reduction of irrelevant features and graph clustering.

Yu and Liu [6] proposed a hybrid FCBF to find the most appropriate optimal discriminative feature subset by trying to remove redundancy in features. Song et al. [27] have proposed FAST for multidimensional data. The algorithm is a little different because it operates in two stages. The first stage divides the features into clusters using graph theory and the second stage selects the most informative features that are closely related to the target class in each cluster to create a subset of final features.

Wrapper Methods

They use a subset of features to train models. Based on a previously generated model, features are added or removed from the selected subset. The problem is thus substantially reduced to a search problem as shown in Figure 5. The only limitation is that the method is computationally expensive. Some examples include forward feature selection, backward feature elimination, and recursive feature elimination. The recursive feature elimination algorithm, is an example from this category [30].

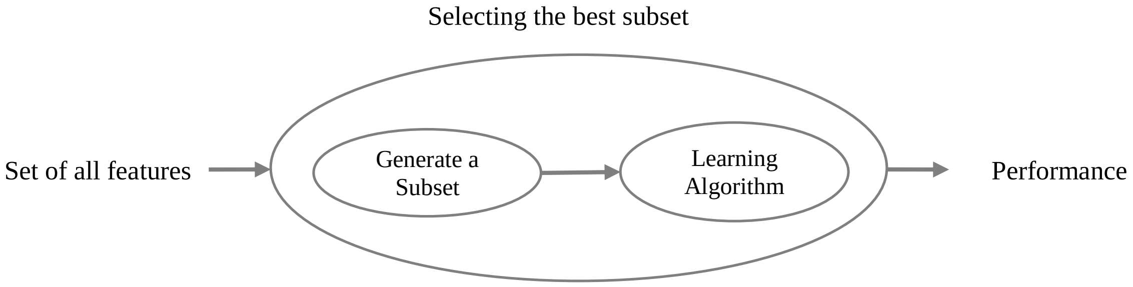

Figure 5: Wrapper method process

Recursive feature elimination [31] selects features by selecting smaller sets recursively according to the features. The first step is to train an estimator from an initial set of features. This is to develop a deep learning on the importance of each feature. This process is conducted recursively and pruned till the desired number of features is achieved.

Embedded Methods

These methods benefit from the qualities of filter and wrapper methods combined. They are implemented using algorithms with inbuilt feature selection methods. They are based on learning about which feature contributes the most to the accuracy of the model as it is being created as shown in Figure 6. Embedded methods have three types: pruning methods, models with inbuilt mechanisms for feature selection and regularization models.

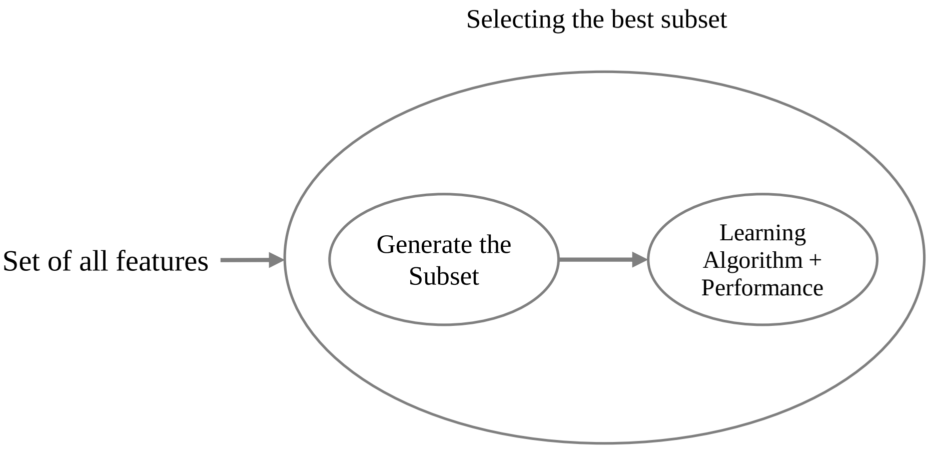

Figure 6: The process for an embedded method

Pruning means selecting a subtree that leads to the lowest test error rate. It begins by using all the available features to train a model. Then, eliminates the features by setting the value as zero of the corresponding coefficients without reducing the performance. These methods use models such as recursive feature elimination with a support vector machine (SVM) [31] which is a supervised machine learning algorithm that can be used for both classification and regression challenges.

Models with inbuilt mechanisms for feature selection include ID3 [31] and C4.5 [31]. The ID3 [31] iterative dichotomizer was the first of three decision tree implementations that Ross Quinlan developed. ID3 builds a decision tree for the given data in a top down fashion starting from a set of objects. C4.5 [31] is an improved version of Quinlan's earlier ID3 algorithm and is used to generate a classification decision tree from a set of training data (in the same way as in ID3) using the concept of information entropy.

Regularization models rely mostly on objective functions to reduce fitting errors to the lowest. It also aims to force the coefficients to be small and potentially reaching zero in the meantime. Due to the good performance of regularization models, researchers have made more efforts in this area. Famous algorithms from this category include lasso [32][33] and elastic net [34].

Lasso [32][33] is a method of regression analysis performing both the tasks of selecting a variable and regularizing. This improves the prediction accuracy and interpretability of the statistical model. Tibshirani [32] introduced this method, which is based on Leo Breiman's nonnegative garrote.

Elastic net regularization [34] is an improved version of lasso [32][33]. It improves the performance of regression analysis models of Lasso by penalizing for additional regression in case there are more predictors than the sample size. This leads to improvements in prediction accuracy by allowing the methods to select only the strongly correlated variables.

2.1.2 Structural Features

This section provides a review of feature selection algorithms for structured features. These features are treated like groups that have some regulatory relationships. These structural features include graph, group and tree structures [35].

Graph Features

A graph is a set of objects in which some pairs of objects are connected by links. Let $\mathcal{G} = (N, E)$ be a given graph, where $N = (1, 2, \ldots\ldots, m)$ is a set of nodes and a set of edges E. Node i is equivalent to the ith feature, and $\mathbf{A} \in \mathbb{R}^{\boldsymbol{m\,x\,m}}$ is used to donate the adjacent matrix of $\mathcal{G}$. Thus, the nodes are representative of the features and the edges represent the relationships between those features [35]. A real application of this category is natural language processing. An instance of this is WordNet. It could indicate the words that are synonyms or antonyms. There is evidence in biological studies that genes work in groups based on their biological functions. Some regulatory relationships have been found among those genes. Three typical algorithms are Laplacian lasso [36], graph-guided fused lasso (GFLasso) [37] and GOSCAR [38].

In a Laplacian lasso [36] features show graph structures. When two features are connected by an edge, chances are that they will be selected together. Therefore, they will show matching feature coefficients. This can be achieved via a graph lasso by adding a graph regularization to the feature graphs on the basis of the lasso method.

Graph-guided fused lasso (GFLasso) [37] is also a lasso variant. It was created to solve the limitations found in the original technique. GFLasso considers positive and negative feature correlations combined explicitly. The limiting factor for GFLasso is the use of pairwise sample correlations for measuring feature dependencies. It is a choice that leads to an added estimation bias. In a small sample size, GFLasso restricts the correct estimation of feature dependencies.

GOSCAR [38] was created to resolve the problems encountered in GFLasso [37] by forcing pairwise feature coefficients to be equal if they were connected over the feature graph.

Group Structure

The group structure is about extracting highly informative subgraphs from a set of graphs. However, some criterion of filtering must be applied. The frequency of sub-graph is a commonly used method. An application of this category in the real world can be found in speed and signal processing. Here, groups can represent the various frequency bands. Two typical algorithms are group lasso [39] and sparse group lasso [40].

Group lasso [39] provides for a combined selection of covariates as a single unit. In this case, it proves quite beneficial. One of the applications of this technique is in performing group selections or selecting group subsets. If a group is chosen, it means that all the contained features are selected as well.

Sparse group lasso [40] has the added ability to choose groups and features in the selected groups in parallel.

Tree Structure

In a tree structure, the features are used to simulate a hierarchical tree with a root value and subtrees (children of parent nodes). It is represented as a set of linked nodes. A real application of this category is in image processing. In image processing, a tree structure could be used to represent the pixels from an image with a face in it. The parent node holds the information of series of child nodes of the image describing spatial locality. Genes and proteins in biological studies can form a certain tree structure according to hierarchy.

The typical algorithm in this structure is a guided tree group lasso [41]. It was proposed for handling feature selection represented in the form of an index tree. In a tree-guided group lasso, the structure of the features can be shown as a tree and the leaf nodes are the features. The internal nodes represent the group of features in a way that each internal node is taken as a root of a subtree and all the features that are grouped are the leaf nodes. Every internal node is assigned a weight and height of that subtree which indicates the tightness of features of that subtree.

2.2 Streaming Feature Selection

A preliminary distinction is needed between streaming data and streaming features. For streaming data, the total number of features is fixed [42]. Also, candidate instances in streaming data are generated dynamically if the size of the instances is unknown. On the other hand, streaming features are the opposite case since the number of instances is fixed. However, the candidate features are generated dynamically if the size of the features is unknown. Streaming feature selection has practical significance in many applications. For example, users of the famous microblogging website Twitter produce more than 250 million tweets per day, including many new words and abbreviations (i.e., features). In the case of tweets, performing feature selection is not recommended due to longer wait time until all the features are generated. Therefore, the use of streaming feature selection is preferred. Figure 7 presents a basic framework for this method.

Step 1: Populate a new feature from the feature stream.

Step 2: Determine whether adding the new feature to the
selected feature set is needed.

Step 3: Update the exiting feature set.

Step 4: Repeat Steps 1 through 3.

The algorithm could have diverse implementations for Steps 2 and 3. In some studies [43]–[46], Step 3 is considered an optional step in which only some of the streaming feature selection algorithm from Step 2 is implemented.

The benefit of this framework selection is in its ability to find an optimal subset. This framework avoids implicitly handling feature redundancy and efficiently eliminates features that are not required by explicitly managing redundancy found in the features [6].

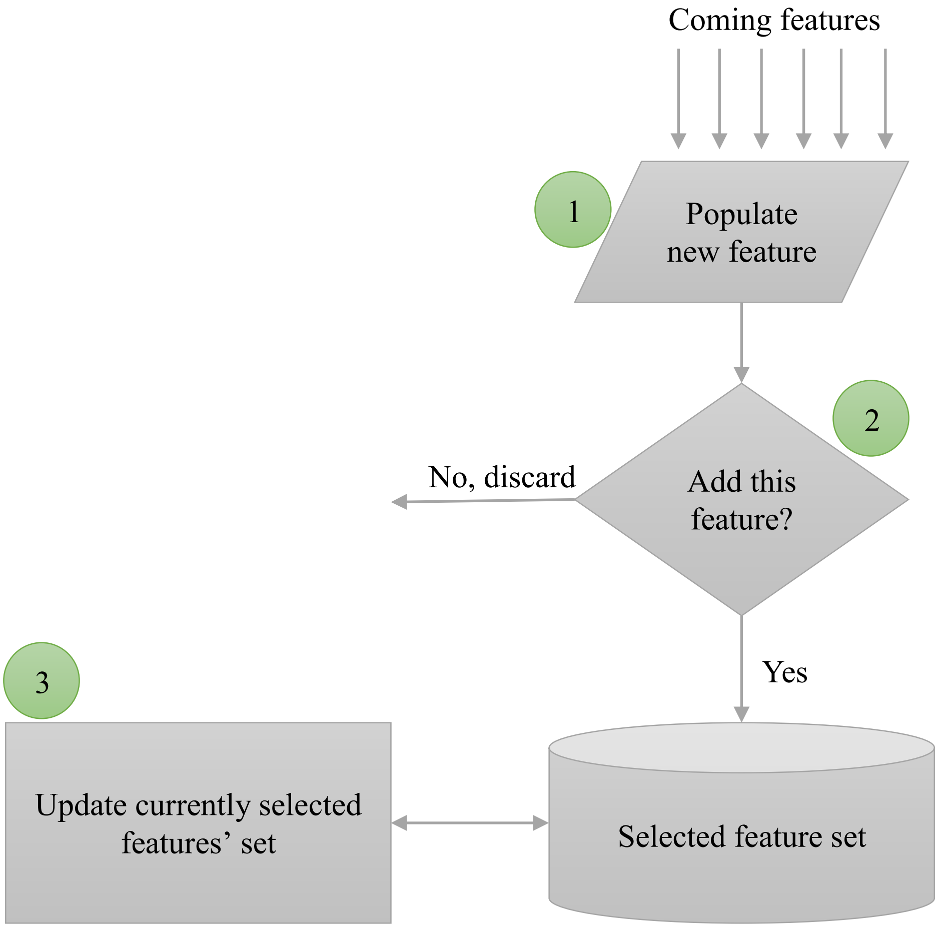

Figure 7: General framework for streaming feature selection

2.2.1 Online Streaming Feature Selection – Single Selection

IBM [47] defined big data analytics as the use of techniques that can handle datasets from large and diverse backgrounds and multiple types. It does not matter whether it is structured and unstructured or streaming and varies according to sizes. Performing feature selection to lower data dimensionality is the desired phase in big data analytics. This phase comes before prediction.

Grafting [43] was considered as the first attempt towards streaming feature selection. It was proposed in 2003 by Perkins and Theiler. Grafting is a popular framework for streaming feature selection and regarded as a general technique for application in a variety of parameterized models using a weight vector $\boldsymbol{w}$ that is subject to $\ell 1$ regularization. The variables in the proposed algorithms are considered one at a time. The weights are re-optimized according to the available set of variables. The tasks in Perkins and Theiler’s study were to select the feature subset and return the corresponding model for every unit time step. According to [43], there were uncertainties in the performance of feature selection methods in this situation. They provided an alternative method known as grafting which was a stage-wise technique for gradient descent.

In 2006, Zhou et al. [44] proposed alpha investing, another of the earliest representative online feature selection approaches (along with grafting [43]). Alpha investing or α investing used p values rather than information theory. In the case of a p-value linked with t-statistic, it is the probability that coefficients of observed sizes can be estimated through chance, even in the event of the true coefficient being zero.

The aim behind alpha investing was to control the threshold during feature selection. This was made possible by selecting new features in the model. Alpha was “invested” thereby increasing the wealth and threshold and allowing for a slight increase in the inclusion of incorrect features in future. In every instance when a feature is tested and determined to be insignificant, wealth is “spent” which reduces the threshold [35]. In the case of alpha investing method, it sequentially acknowledges newer features for feeding into a predictive model and modeling the set of candidate features in the form of a dynamically generated stream. One of the benefits of using alpha investing is its ability to handle feature sets of unknown sizes even up to infinity. The use of linear and logistic regression to dynamically adjust the reduction threshold for errors is favored such that the predictive model needs to evaluate a new feature for inclusion for each instance.

In another study Wu et al. [45] used information theory to find the answer to streaming feature selection by utilizing Markov blanket concept. In earlier studies, Wu et al. developed a framework that used feature relevance and a new algorithm called as OSFS along with its novel adaptation called as Fast-OSFS. According to the published definitions in the study, the features could be classified into one of these four categories: irrelevant features, redundant features, weakly relevant but non-redundant features and strongly relevant features. Thus, OSFS finds its application in online selection for features that are non-redundant and strongly relevant using two step method. The first step is an analysis of its online relevance and second is online redundancy analysis. Furthermore, Wu et al. [48] described the working of a Fast-OSFS algorithm that improves the efficiency of OSFS. The concept behind Fast-OSFS is the breakup of online redundancy analysis into two steps a) inner-redundancy analysis and b) outer-redundancy analysis. Additionally, the same authors published an updated study [48] in which they introduced an efficient Fast-OSFS algorithm that improved the performance of streaming feature selection. The algorithm proposed in this study was evaluated on a large scale using multidimensional datasets.

Yu et al. [49] proposed another approach known as scalable and accurate online approach (SAOLA) for handling multidimensional datasets feature selection sequentially. SAOLA is based on a theoretical analysis and derived from a low bound of correlations between features for pairwise comparisons. It was followed by a set of pairwise online comparisons for maintaining the parsimonious online model over longer durations.

Eskandari and Javidi [46] proposed a new algorithm called OS-NRRSAR-SA algorithm to resolve OSFS from the rough sets (RS) perspective. This algorithm adopts the classical concept of RS based feature significance to reduce non-relevant features. Eskandari and Javidi claimed that the primary advantage of the algorithm was that it did not need prior domain knowledge concerning the feature space making it a viable alternative for true OSFS scenarios.

Wang et al. [50] proposed the dimension incremental algorithm for reduction computation (DIA-RED). This algorithm maintained the RS-based entropy value of the currently selected subsets and updated that value whenever new conditional features were added. While DIA-RED is capable of handling streaming scenarios despite having limited or no knowledge of the feature space, it can manage with the information contained in the lower approximation of a set and avoid using information contained in the boundary region. Therefore, real-value datasets cannot benefit from this algorithm. Also, DIA-RED algorithm does not possess an effective mechanism that eliminates redundant attributes which leads to the generation of large subsets during feature streaming. This is a prime reason for ineffective partitioning and at the time of calculating RS approximations. Therefore, the algorithm falls short of its expectations in handling most real-world datasets.

Gangurde [28] and Gangurde and Metre [29] proposed a novel clustering concept to manage big data dimensionality reduction problem. A minimum spanning tree was used to reduce the complexity in calculating feature selection and obtain a formatting of clusters. However, this concept's work scope is limited to dimensionality reduction.

Javidi and Eskandari [51] have proposed a method that employs significance analysis concept in the theory of rough sets for controlling unknown feature space in SFS problems. The primary motivation for their consideration was that RS-based mining of data hardly used any domain knowledge besides the datasets that were provided. The algorithm was evaluated using several multidimensional datasets for its compactness, running time and classification accuracy.

Tommasel and Godoy [52] presented an online feature selection method for multidimensional data that is dependent on the combination of social and contextual information. The goal of their work was classifying short texts that are generated simultaneously in social networks.

Zhou et al. [53] proposed an online streaming feature selection method using adaptive density neighborhood relation, called OFS-Density. They claimed that their approach has not required domain information before learning. OFS-Density used the density information of the surrounding instances, which did not need to specify any parameters in advance. Depending on the fuzzy equal constraint, OFS-Density could choose features with low redundancy.

2.2.2 Online Streaming Feature Selection – Group Selection

Li et al. [54] proposes the group feature selection with streaming feature (GFSSF) at both levels – individual and group as a feature stream instead of a predefined feature set. Wu et al. also illustrated the GFSSF algorithm, which is segregated into two distinct levels of selection. The first one at the feature level and second at the group level is based on the tenets of information theory. Features from the same group are processed in the case of feature level selection. Redundancy analysis is used for selecting the best feature subset from the features that have arrived so far. In contrast, a set of feature groups were reviewed to cover the uncertainty to a large extent in the class labels at a minimum cost during the group level selection phase. Later on, this method finds a subset of features that seem relevant and are sparse in both individual and group feature levels. In work done to date, single features are being targeted primarily and group features are left unaddressed. Information theory is being used only for recognizing irrelevant features.

In 2015, Yu et al. [55] extended SAOLA, their previous method [49] to handle a type of online streaming group feature selection and called this group-SAOLA. The new group-SAOLA algorithm could maintain an online set of feature groups that are sparse at the group feature level as well as individual feature levels at the same time. For the group level, Yu et al. claimed that the group-SAOLA algorithm, while online could generate a set of feature groups that is sparse both between groups and within each group. This would maximize the methods of predictive performance in classification.

Wang et al. [56][57] tried to handle both single and group streaming feature selection by introducing an online group feature selection (OGFS) algorithm for image classification and face verification. Wang et al. divided online group feature selection into the online intragroup selection and intergroup selection. They designed two criteria for intragroup selection based on spectral analysis and introduced the lasso algorithm to reduce the redundancy in intergroup selections.

2.2.3 Feature Grouping

García-Torres et al. [26] proposed a feature selection strategy that utilized feature grouping to increase the effectiveness of the feature search termed the "predominant group-based variable neighborhood search" (PGVNS). PGVNS is based on the concepts of an approximate Markov blanket and a predominant feature. In their work, they introduced the idea of a predominant group and proposed a heuristic strategy called GreedyPGG for grouping the input space. In their experiments, they used synthetic and real datasets from the microarray and text mining domains for testing the PGVNS, and they compared the result with those of three popular feature-selection algorithms: Fast Correlation-Based Filter (FCBF) [6], Fast clustering-based feature selection algorithm (FAST) [27], and CVNS [26]. It is planned to use this work as a baseline for this dissertation work. However, it is aimed to develop a unique approach, which handles the streaming feature selection.

Gangurde et al. [28][29] in their two published papers proposed a clustering concept for feature selection to handle the reduction of big data dimensionality. The formatting of clusters was obtained from a minimum spanning tree that reduced the complexity of the computation of feature selection. This work is more about graph clustering and reducing the irrelevant features. In this dissertation work, it is planned to reduce the irrelevant and redundant features.

Yu and Liu [6] proposed a Fast Correlation-Based Filter (FCBF) which is a hybrid technique to search the optimal discriminative feature subset by considering removing feature redundancy. As mentioned previously Song et al. [27] proposed a fast clustering-based feature subset selection algorithm (FAST) for high-dimensional data. The algorithm works on two steps: Firstly, features are divided into clusters by using graph-theoretic clustering methods. Secondly, the most informative feature that is strongly related to target classes is selected from each cluster to form a subset of final features.

2.3 Application of Streaming Feature Selection

Yu et al. [58] developed the first comprehensive open-source library, called LOFS, for use in MATLAB and OCTAVE that implemented the state-of-the-art algorithms of online streaming feature selection. The library was designed to facilitate the development of new algorithms in this research direction and made comparisons between the new methods and the existing ones. The learning module consisted of two submodules: (1) learning features added Individually (LFI) and (2) learning grouped features added sequentially (LGF).

Zhuang et al. [59] applied four state-of-the-art online streaming feature selection methods to build long-lead extreme floods forecasting models. The methods were: (1) alpha-investing, (2) OSFS, (3) SAOLA, and (4) group SAOLA. The use of these four algorithms allowed them to get the benefit of big data analytics to successfully estimate what was expected to happen in the future for both flood information management and long-lead extreme flood forecasting. The prediction models were evaluated and compared systematically to the historical precipitation and associated meteorological data collected in the state of Iowa.

2.4 Attribute Evaluation Relevancy and Feature Redundancy

The objective of streaming feature selection is to choose (while online) the subset of features from a multidimensional data which leads to an increase in accuracy and robustness. This can be achieved by removing the features that are irrelevant and redundant.

In streaming feature selection, the optimal, final feature subset should be relevant to the class and should not be redundant with any other existing features to increase robustness. Thus, it could determine two feature testing stages that would be used in selecting the final and most optimal subset. Thus, it could use relevance analysis which can determine the subset of relevant features while removing the irrelevant ones. Similarly, it could use redundancy analysis to remove redundant features and leave a final subset as depicted in Figure 8.

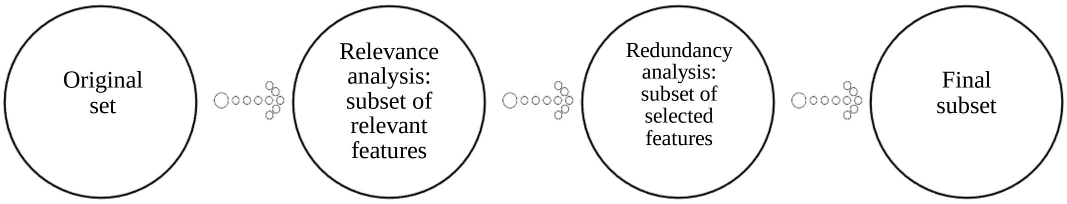
Original set
Relevance analysis: subset of relevant features
Redundancy analysis: subset of selected features
Final subset

Figure 8: Relevancy and redundancy evaluation

2.4.1 Relevance Analysis

In relevance analysis, a single feature’s relevance to the selected class is evaluated. The criterion for relevancy decides how effectively a variable can distinguish between a class or a feature and a class [60].

$$Relevance\ Test\ (X, Y) = how\ useful\ X\ is\ for\ predicting\ Y \qquad (5)$$

In feature relevancy, a feature is evaluated individually and discarded if it fails to reach a chosen cutoff point. Table 2 is a comparison of some existing algorithms that are used to evaluate a feature’s relevancy to a class as part of a classification problem.

Chi-squared [61] is used to calculate the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.

Gain ratio (GR) [61] is used to evaluate the worth of an attribute by measuring the gain ratio with respect to the class. The gain ratio is given by

$$GR = \frac{H(class) - H(class|attribute)}{H(attribute)} \tag{6}$$

where H is the entropy.

Information gain (IG) [61] is used to evaluate an attribute’s worth by measuring the information gain with respect to the class. The information gain is given by

$$IG = H(class) - H(class\ |attribute)\ . \tag{7}$$

ReliefF [61] is used to evaluate an attribute’s worth by sampling an instance several times and taking the value of the given attribute for the nearest instance of the same class and of a different class. The formula for ReliefF is

$$W(A_l) = W(A_l) - \frac{\sum_{j=1}^{k} diff(A_l, R_i, H_j)}{g*k} + \frac{\sum_{c \neq class(R_i)}[\frac{p(c)}{1-p(class(R_i))} \sum_{j=1}^{k} diff(A_l, R_i, M_j(c))]}{g*k}, \tag{8}$$

where

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value\ (A, I_2)|}{\max(A) - \min(A)}\ . \tag{9}$$

Significance [61] is used to evaluate an attribute's worth by computing its probabilistic significance as a two-way function (both attribute-class and class-attribute associations).

Symmetrical uncertainty (SU) [61] is used to evaluate an attribute's worth by measuring its symmetrical uncertainty with respect to a class; it is given by

$$SU = 2 * \frac{H(class) - H\ (class\ |attribute)}{H(class) + H(attribute)}\ . \tag{10}$$

2.4.2 Redundancy Analysis

Redundancy analysis is used to evaluate the features' similarity. In other words, it is used to answer the question: How much can adding a new feature to improve the accuracy of a machine-learning model?

Yu and Liu [12] defined a feature as predominant (both relevant and non-redundant) if it does not have an approximate Markov blanket in the current set. For two relevant features, $F_i\ and\ F_j\ (i \neq j), F_j$ forms an approximate Markov blanket for F_i if

$$SU_{j,c}\ \geq\ SU_{i,c}\ and\ SU_{i,j}\ \geq\ SU_{i,c}\ , \tag{11}$$

where $SU_{j,c}$ is a correlation between any feature and class and $SU_{i,j}$ is a correlation between any pair of features, F_i if and F_j $(i \neq j)$.

Correlation-based feature selection (CFS) [62][61] is a popular technique for ranking the relevancy of features by measuring the correlations between features and classes and between features and other features.

Given k features and C classes, CFS defines the relevancy of the feature subset using Pearson's correlation equation:

$$Merit_s = \frac{kr_{kc}}{\sqrt{k+(k-1)r_{kk}}}, \tag{12}$$

where $Merit_s$ is the relevancy of the feature subset, r_{kc} which is defined as the average linear correlation coefficient among features and classes. Also, r_{kk} is defined as the average linear correlation coefficient among unique individual features. Normally, CFS adds or deletes one feature at a time using forward or backward selection. However, this research used a sequential forward floating search (SFFS) as the search direction.

Sequential forward floating search (SFFS) [61][63] is a classic heuristic searching method. It is a variation of bidirectional search and sequential forward search and is thus part of the dominant direction of forward search. SFFS removes features (backward elimination) after adding features (forward selection).

The numbers of forward and backward steps are not fixed and can be controlled dynamically depending on the criterion of the selected subset. This eliminates the need for parameter setting.

2.5 Streaming Feature Selection with Big Data Challenges

As mentioned earlier, big data has created challenges that are yet to be addressed by traditional machine learning practices. This has led to the adoption of methodologies capable of handling increasingly large data volumes. To overcome this challenge, improving streaming feature selection is necessary to introduce better and more efficient approaches for handling extremely high dimensionality of big data. This section highlights some of these challenges which could be considered hot topics in streaming feature selection.

2.5.1 Scalability

Scalability is defined as “the impact of an increase in the size of the training set on the computational performance of an algorithm in terms of accuracy, training time and allocated memory” [64]. Today, with the exposure of big data, those who use traditional methods are struggling to cope with the extreme high-dimensionality of big data as they attempt to extract satisfactory results in a reasonable time.

The extremely multidimensional big data is unable to load in the memory in a single data scan. Therefore, it is challenging to get a score of feature relevancy without considering sufficient density surrounding every sample.

Considering the available approaches for large-scale selection of features there are two prominent phases. The first phase measures the relevancy of individual features and then ranks them according to their relevance values. The values that show the highest relevancy only are used for input in the second phase. However, this approach presents the limitations that it may remove the features that are lowly ranked or even considers its interactions with other features [65].

2.5.2 Stability

The stability of feature selection is defined [42] as the sensitivity that the selection process has to data perturbation in the training set. Stability quantifies how a training set affects feature selection. The feature selection algorithm for classification is measured using classification accuracy. Thus, the stability of any algorithm is a critical factor when developing feature selection.

Alelyani et al. [66] has presented and argued for some characteristics of data that may play a vital role in stabilizing the algorithm. They are dimensionality (m), size of the sample (n) and data distribution across folds. Therefore, the stability issue tends to be dependent on data.

A measure of stability requires a similarity measure for feature preferences. Researchers have proposed various stability measures to evaluate robustness [67][68][64]. These measures can be placed in three categories:

Category 1: A weight or score is assigned to each feature, indicating its importance. For a vector of features$f = (f_1, f_2, \ldots, f_m)$, this category produces a feature set as follows:

$weighting - scoring: w = (w_1, w_2, \ldots, w_m), w \in W \subseteq R^m$.

Category 2: This is a simplification of the first category; ranks are assigned to features instead of weights. For a vector of features $f = (f_1, f_2, \ldots, f_m)$, this category produces a feature set as: follows

$ranking: r = (r_1, r_2, \ldots, r_m), 1 \leq r_i \leq m$.

Category 3: These measures consist of sets of selected features for which no weighting or ranking is considered. For a vector of features $f = (f_1, f_2, \ldots, f_m)$, this category produces a feature set as follows:

$subset\ of\ features: s = (s_1, s_2, \ldots, s_m),\ s_i \in \{0,1\}$, with 0 indicating the absence of a feature and 1 for presence.

For streaming feature selection, the challenge lies with the unknown features. Selecting the most informative features from among the current features challenges the stability of any proposed algorithm. As a result, updating the selected subset also challenges the robustness of the algorithm.

2.5.3 Sustainability

The volume of data increases by 90% of the data in the world which has been created in the last two years [69]. Data is generated from different resources like mobile phones, sensors, and social media in continuous manner. This data is expected to grow soon dramatically. The data revolution would pose a challenge for resources sustainability. Sustainability means the ability to optimize resource usage. Thus, finding a new way to reduce the extremely high dimensionality of big data would result in significant savings in the analytic process. It is clear from previous examples that feature selection would be considered as the first option to reduce the dimensionality of any data. This would allow picking informative features only rather than considering all of them. Consequently, the streaming feature selection would efficiently resolve the sustainability issue of streaming big data. Recently [70][69][71][72] highlight has been the greening issue of big data analytics. The process of big data analytics is accompanied with a lot of computing workloads, which is time consuming at the same time energy and resource demanding.

2.6 Discussion and Comparison

This section discusses streaming feature selection algorithms and examples that are demonstrated in Section 2.2. It also compares these algorithms based on big data challenges that were discussed in Section 2.5. Table 2 is a comparison of the reviewed streaming feature selection algorithms. Note that these algorithms use either single feature selection, group feature selection or both. Table 2 presents a comparison of the algorithms based on the feature selection type, how they compare to other online feature selection methods, datasets and classifiers that were used to report the classification accuracy and the environment of the experiment.

As mentioned earlier, grafting [43] and alpha investing [44] are two of the earliest methods for online feature selection. Grafting algorithm is based on a stage wise gradient descent approach for streaming feature selection. However, grafting has some limitations. It can obtain a global optimum with respect to features included in the model, it is not optimal as some features are dropped during online selection. Besides, the gradient retesting over all the selected features greatly increases the total time cost. Thus, tuning a good value for the critical regularization parameter λ requires the information of the global feature space. Similarly, Alpha-investing does not reevaluate the selected features, it hence performs efficiently, but it is probably performing ineffectively in the subsequent feature selection for never evaluating the redundancy of selected features [56]. These limitations for high-dimensional data were recognized at the time they were created. For example, the Pima Indian Diabetes dataset [73] found that grafting has 768 instances and eight attributes. Likewise, alpha investing used a spam dataset [74], which had 4,601 instances and 57 attributes. Wang et al. in their OGFS experiments [56][57], used the method of grafting for performing feature selection using the gradient descent technique which can be quite effective in pixel classification.

However, this method still requires a global feature space for defining key parameters during the selection of features. Therefore, it presents limitations in cases where feature stream is infinite or has an unknown size. Also, alpha investing calculates each new feature using a p-value that is from a regression model. In case where the p-value of a new feature goes to a certain limit or threshold (known as α), the algorithm selects the feature. Therefore, alpha investing never discards a feature once it has been selected.

Currently, researchers focusing on OSFS, Fast-OSFS [45], SAOLA [49] and group-SAOLA [55] are taking the lead in this area. Following their work history, these researchers started with the OSFS [45], Fast-OSFS [45], and SAOLA [49] to handle single feature selection. After that, they introduced group-SAOLA [55] to handle both single and group feature selection. In OSFS [45] features are selected according to the relevance they have online and whether they are redundant or not. Based on the relevance it holds to the class label, input features are labeled as strongly relevant, weakly relevant or non-relevant. Online relevance analysis is used to remove irrelevant features. Markov blankets are used to remove redundant features. In the case of OSFS, every time a method includes a new feature, it is necessary to reanalyze the redundancy of all selected features. To improve the performance of conducting redundancy analysis, a fast-version of OSFS is proposed known as Fast-OSFS [45]. The Fast-OSFS experiments uses eight UCI [75] benchmark databases. Researchers compared Fast-OSFS's performance with those of grafting and alpha investing [76] algorithms using the k-nearest neighbor (or k-nn), decision tree, and random forest datasets. SAOLA managed to handle a multidimensional dataset which allowed it to overcome the two challenges of big data – scalability and extreme multidimensionality.

Another attempt to resolve the problem of streaming feature selection is OS-NRRSAR-SA [46]. This method uses RS-based data mining to control unknown feature space without needing any domain knowledge. During experiments, Eskandari and Javidi compared the algorithm's performance with those of four modern algorithms (grafting, information investing [76], fast-OSFS, and DIA-RED) using 14 benchmark datasets. For these experiments, the computer had 24 GB of memory which gave this algorithm a performance benefit relative to other algorithms.

DIA-RED [50], another single feature selection algorithm was proposed to resolve the issue of streaming feature selection. In the experiments on this method, the researchers used only six datasets from UCI's [75] repository of machine learning: Backup-large, Dermatology, Splice, Kr-vs-kp, Mushroom, and Ticdata2000. However, the researchers didn't compare their method to other state-of-art streaming-feature-selection algorithms. They only measured the uncertainty of the tested datasets compared to the traditional feature selection approaches.

On the other hand, GFSSF [54], group-SAOLA [55] and OGFS [56][57] were designed to handle group feature selection. The GFSSF algorithm has the edge over both group-SAOLA [55] and OGFS [56][57] according to a comparison with lasso [39] which is a group feature selection algorithm. However, in terms of big data, group-SAOLA used fewer resources such as memory. Using more resources would enhance these methods chance of prevailing in the big data scalability challenge. Table 3 contains a comparison of some of the reviewed streaming feature selection algorithms. This comparison is based on the approach used to reduce the redundancy of the received features.

Table 2: Properties of the experiments on streaming feature selection

Algorithm	**Properties**
Grafting [43]	• *Single or group feature selection:* single. • *Compared with which algorithms*: none. • *Datasets:* Two synthetic datasets (A and B) and Pima Indian Diabetes dataset (Blake & Merz, 1998) [73]. • *Classifiers:* Combination of the speed of filters and the accuracy of the wrapper. • *Environment:* Not mentioned.
Alpha investing [44]	• *Single or group feature selection:* single. • *Compared with which algorithms*: none. The appraisal was limited to the accuracy of the whole dataset. • *Datasets:* Seven datasets from the UCI [75] repository: cleve, internet, ionosphere, spam, spect, wdbc, and wpbc. Three datasets on gene expression: aml, ha, and hung. • *Classifiers:* C4.5, fivefold cross-validation. • *Environment:* Not mentioned.
OSFS and Fast-OSFS [45]	• *Single or group feature selection:* single. • *Compared with which algorithms*: Grafting and alpha investing [76]. • *Datasets:* Ten public challenge datasets: lymphoma, ovarian-cancer, breast-cancer, hiva, nova, manelon, arcene, dexter, dorohthea and sido0. • *Classifiers:* k-nn, decision tree (J48) and random forest (Spider 2010). • *Environment:* Windows XP, a 2.6 GHz CPU, and 2 GB memory.
SAOLA [49]	• *Single or group feature selection:* single. • *Compared with which algorithms*: Fast-OSFS [48], alpha investing [76], OFS [77], FCBF [6], as well as two state-of-the-art algorithms, SPSF-LAR [78] and GDM [79]. • *Datasets:* Ten high-dimensional datasets: two public microarray datasets (lung cancer and leukemia), two text-categorization datasets (ohsumed and apcj etiology), two biomedical datasets (hiva and breast cancer), three NIPS 2003 (dexter, madelon, and dorothea) and the thrombin dataset, which was chosen from KDD Cup 2001. Four extremely high-dimensional datasets from the Libsvm dataset website: news20, url1, webspam, and kdd2010. • *Classifiers:* KNN and J48, which are provided in the Spider Toolbox2 [80]. • *Environment:* Intel i7-2600 with a 3.4 GHz CPU and 24 GB of memory.

Table 2: Properties of the experiments on streaming feature selection (continued)

Algorithm	**Properties**
OS-NRRSAR-SA [46]	• *Single or group feature selection:* single. • *Compared with which algorithms*: Grafting, information investing [76], fast-OSFS, and DIA-RED. • *Datasets:* Fourteen high-dimensional datasets: The dorothea, arcene, dexter, and madelon datasets from the NIPS 2003 Feature-Selection Challenge. The nova, sylva, and hiva datasets from the WCCI 2006 Performance Prediction Challenges. The sido0 and cina0 datasets from the WCCI 2008 Causation and Prediction Challenges. The arrhythmia and multiple features datasets from the UCI Machine Learning Repository. Three synthetic datasets: tm1, tm2, and tm3. • *Classifiers:* J48, JRip, Naive Bayes, and kernel SVM with the RBF kernel function. • *Environment:* Dell workstation with Windows 7, 2 GB of memory, and a 2.4 GHz CPU.
DIA-RED [50]	• *Single or group feature selection:* single. • *Compared with which algorithms*: None. • *Datasets:* Six datasets from the UCI [75] Machine-Learning Repository: Backup-large, Dermatology, Splice, Kr-vs-kp, Mushroom, and Ticdata2000. • *Classifiers:* information entropy used to measure the uncertainty of a dataset: complementary entropy [81], combination entropy [82], and Shannon's entropy [83]. • *Environment:* Windows 7, an Intel Core i7-2600 CPU (2.66 GHz), and 4 GB of memory.
GFSSF [54]	• *Single or group feature selection:* single and Group selection. • *Compared with which algorithms*: Five standard feature-selection algorithms: MIFS [17], joint mutual information [84], mRMR [11], ReliefF [24], and lasso [32]. Four streaming-feature-selection algorithms: grafting [43], α investing [44], OSFS [45], and Fast-OSFS [45]. One group-feature-selection algorithm: group lasso [39].

Table 2: Properties of the experiments on streaming feature selection (continued)

Algorithm	**Properties**
GFSSF [54], continued	• *Datasets:* Five UCI [75] benchmark datasets: WDBC, WPBC, IONOSPHERE, SPECTF, and ARRHYTHMIA. Five challenge datasets with relatively high feature dimensions) downloaded from http://mldata.org/repository): DLBCL (7,130 features; 77 instances), LUNG (7,130 features; 96 instances), CNS (7,130 features; 96 instances), ARCENE (10,000 features; 100 instances), and OVARIAN (15,155 features; 253 instances). Five UCI [75] datasets with generated group structures: HILL-VALLEY (400 features; 606 instances), NORTHIX (800 features; 115 instances), MADELON (2,000 features; 4,400 instances), ISOLET (2,468 features; 7,797 instances), and MULTI-FEATURES (2,567 features; 2,000 instances). • *Classifiers:* NaiveBayes [85], k-NN [86], C4.5 [87], and Randomforest [88]. • *Environment:* Windows 7, a 3.33 GHz dual-core CPU, and 4 GB of memory.
group-SAOLA [55]	• *Single or group feature selection:* group • *Compared with which algorithms*: Three state-of-the-art online-feature-selection methods: Fast-OSFS [48], alpha investing [44], and OFS [48]. Three batch methods: one well-established algorithm (FCBF) [6], and two state-of-the-art algorithms (SPSF-LAR [78] and GDM [79]). • *Datasets:* Ten high-dimensional datasets: madelon, hiva, leukemia, lung-cancer, ohsumed, breast-cancer, dexter, apcj-etiology, dorothea, and thrombin. Four extremely high-dimensional datasets: news20, url1, webspam, and kdd2010. • *Classifiers:* KNN and J48, which are provided in the Spider Toolbox [80], and SVM. • *Environment:* Intel i7-2600, a 3.4 GHz CPU, and 24 GB of memory.

Table 2: Properties of the experiments on streaming feature selection (continued)

Algorithm	**Properties**
OGFS [56][57]	• *Single or group feature selection:* single and group. • *Compared with which algorithms*: Grafting, alpha investing, and OSFS. • *Datasets:* Eight datasets from UCI: Wdbc, Ionosphere, Spectf, Spambase, Colon, Prostate, Leukemia and Lungcancer. Three datasets from the real world: Soccer, Flower-17, and 15 Scenes. • *Classifiers:* appraisal was based on number of the selected features. • *Environment:* Windows XP, a 2.5 GHz CPU, and 2 GB of memory.

2.7 Current State-of-Arts Areas vs. The Proposed Approach

Table 3 highlights the current state of the art areas and the new area the proposed work is cover. This comparison is based on the approach used to reduce the redundancy of the received features. The last row in the table shows the areas that the proposed approach is going to cover.

Table 3: Comparison of related works areas and the proposed approach to address all areas

Related work	**Method**			
	Feature selection	**Streaming feature selection**	**Feature grouping**	**Streaming feature grouping and selection**
Grafting [43]	√	√		
Alpha investing [44]	√	√		
PGVNS [26]	√		√	
FCBF [6]	√		√	
OSFS and Fast-OSFS [45]	√	√		
SAOLA [49]	√	√		
OS-NRRSAR-SA [46]	√	√		
DIA-RED [50]	√			
Gangurde [28] and Gangurde and Metre [29]	√		√	
group-SAOLA [55]	√	√		
OGFS [56][57]	√	√		
The proposed approach	√	√	√	√

In the next chapter, the broad range of applications of traditional feature selection in reducing the high dimensionality of streaming data are discussed.

Chapter 3: Traditional Feature Selection and Predicting Patient Deterioration: Study Case

In this chapter, it explores the use of traditional feature selection in reducing the high dimensionality of streaming big data. It uses data collected from the modern intensive care unit (ICU) which have a vast amount of data generated during the patient stay. Each patient would be represented as one instance having 700 attributes. The purpose of this study is exploring how the feature selection could support predicting patient deterioration in the ICU. The last decade has seen considerable advances in the amount of data that is generated and collected in the modern intensive care units (ICUs), as well as the technologies used to analyze and understand it. ICUs are specialist hospital wards, where they provide intensive care (treatment and monitoring) for patients in seriously ill and when their condition changes often. ICUs are considered a critical environment where the decision needs to be carefully taken. This data could be used with the help of intelligent systems, such as data analytics and decision support systems, to determine which patients are at an increased risk of death. Making such decision could allow healthcare professionals to act at early stage. For instance, patients in the ICUs have a wide variety of medical laboratory tests on different body fluids (E.g. blood and urine). The natures of medical lab tests and how often these tests are performed depend on why the patient is in ICU and how stable the patient is.

Medical professionals may order laboratory tests to confirm a diagnosis or monitor patients' health. However, deciding which test is likely to provide further information is a challenge. Recent studies have demonstrated that frequent laboratory testing does not necessarily relate to better outcomes [89].

Dimensionality reduction would be the first solution to eliminate duplicate, useless and irrelevant features. This is typical alternative done while solving machine learning problems to select the most discriminative attributes. This chapter proposes an efficient mining technique to reduce the observation time in ICUs by predicting patient deterioration in its early stages through data analytics. In this dissertation investigation, it study the effect of traditional feature selection on reducing patient deterioration. This can be achieved by selecting the most informative labs' tests. Lab tests are represented by features. First, it use the lab test results to predict patient deterioration. To the best of this dissertation knowledge, this is the first work that primarily uses medical lab tests to predict patient deterioration. Lab test results have a crucial role in medical decision making. Second, it identify the most important medical lab tests using state-of-the-art feature-selection techniques without using any informed domain knowledge. The purpose is to provide reasoned advice at a comparable level to that provided by healthcare experts "consultant". The purpose is to provide reasoned advice at a comparable level to that provided by healthcare experts. In this chapter, it is evaluating the learning model performance in term of feature selection capability without using domain knowledge at this stage. In the future, using domain knowledge to understand how the selected features relate to a health outcome would improve the work.

Finally, the proposed approach helps reduce redundant medical lab tests. Thus, healthcare professionals could identify a subset of the most important intensive care unit (ICU) lab tests that should be fundamental for any patient in the ICU.

ICUs, like other healthcare sectors, are sources of large amount of data that needs analysis. Data mining represents great potential benefits for the ICUs to enable systematically use data and analytics to identify best practices that improve care and reduce costs. Clinical data mining is the application of data mining techniques using clinical data. Data mining with clinical data has three objectives: understanding the clinical data, assisting healthcare professionals, and developing a data analysis methodology suitable for medical data [90].

Data mining is the analysis step of knowledge discovery. It is about the extraction of interesting (non-trivial, implicit, previously unknown, and potentially useful) patterns or knowledge from data [91]. When mining massive datasets, two of the most common, important and immediate problems are sampling and feature selection.

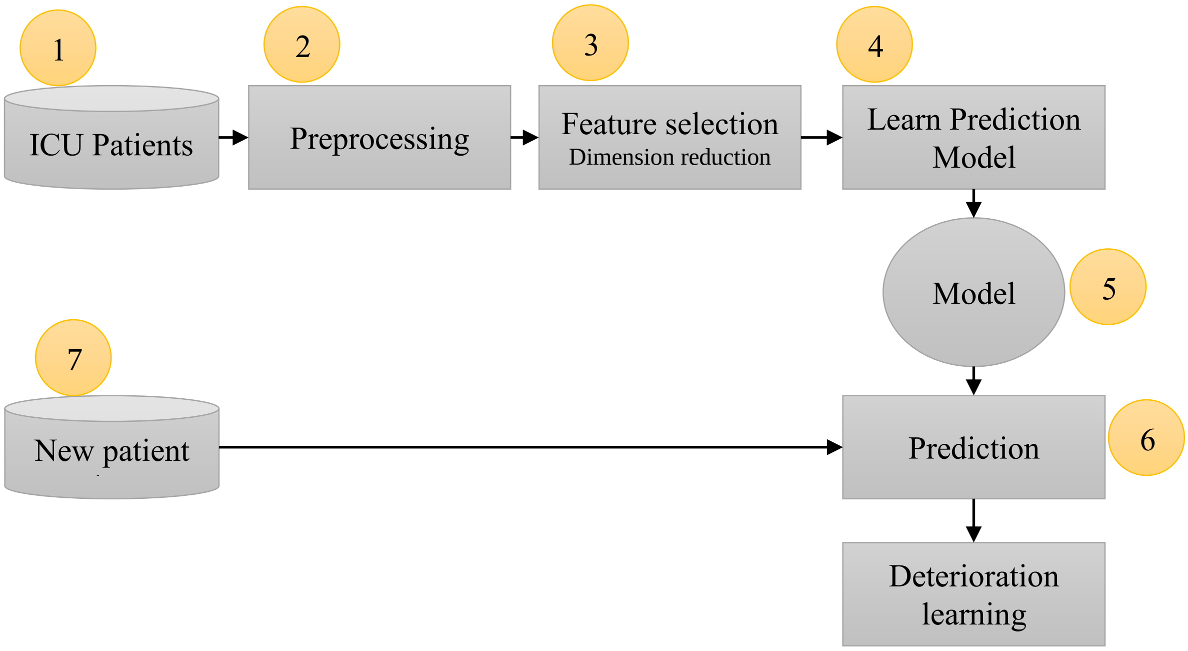

Figure 9: Architecture of the proposed approach

Appropriate sampling and feature selection contribute to selecting the most informative features to obtain satisfactory results in model building [92]. Figure 9 shows the architecture of the proposed technique. The data is collected from the database of ICU patients (step 1). Then the data is integrated, cleaned and relevant features are extracted (step 2). After that, feature selection or dimensionality reduction techniques are applied to obtain the best set of features and reduce the data dimension (step 3). Then the prediction model is learned using a machine learning approach (step 4). When a new patient is admitted to the CPU, the patient's data is collected incrementally (step 5). The patient data is evaluated by the prediction model (step 6) to predict the possibility of deterioration of the patient, and warnings are generated accordingly. In more details, the architecture of the proposed approach is as following:

1) ICU Patient Data: The details of the data and the collection process are discussed in Section 3.1.
2) Preprocessing: At the preprocessing stage, it is used two different datasets. These datasets were generated from a Labevents table. Please refer to Table 4. The first dataset contained the average value of applied medical tests, and the second contained the total number of times each test was applied.
3) Feature Selection / Dimensionality Reduction: attribute selection is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. The goal here is reducing lab tests. Therefore, the medical professional can identify the most important tests to be used in the ICU in order to reduce the redundant tests. This work selects filter methods because they are moderately robust against the overfitting problem, as follows:
    a. Attribute evaluator: InfoGrainAttributeEval
    b. Search method: Ranker

c. Attribute selection mode: use full training set

4) Learning: In this experiment it use a classification technique and five of the most popular classifier techniques: Naïve Bayes classifier, Support vector machine (SVM), ZeroR classifier, decision tree (J48) and RandomForest. Different types of machine learning are used in order to avoid random results.

5) Model: The developed model aims to predict ICU patient deterioration by mining lab test results. Thus, observation time can be reduced in the ICUs and more actions can be taken in the early stages.

6) Prediction: After each new test result, medication event, etc., the patient data is preprocessed, and features are extracted to supply to the prediction model. The model predicts the probability of deterioration for the patient. This probability may change when new data (e.g. more test results) are accumulated and applied to the model. When the deterioration probability reaches a certain threshold specified by the healthcare providers, a warning is generated. This would help the healthcare providers to take proactive measures to save the patient from getting into a critical or fatal condition.

7) New patient data: When a new patient is admitted to the ICU, all his information is stored in the database. Some of these are incremental, such as vital sign readings, lab test results, medication events, and the like. The data of the patient again go through the preprocessing and feature extraction phases before they can be applied to the model.

3.1 MIMIC II Database

The MIMIC-II database is a part of the Multiparameter Intelligent Monitoring in Intensive Care project funded by the National Institute of Biomedical Imaging and Bioengineering at the Laboratory of Computational Physiology at MIT, which was collected from 2001 to 2008 and represented 26,870 adult hospital admissions. In this work, MIMIC-II version 2.6 is used, because it is more stable than the newer version 3, which is still in the beta phase and needs further work of cleaning, optimizing and testing. MIMIC-II consists of two major components: clinical data and physiological waveforms.

The MIMIC dataset has three main features: (1) it is public; (2) it has a diverse and a massive population of ICU patients; and (3) it contains high temporal resolution data, including lab results, electronic documentation, and bedside monitor trends and waveforms [93]. Several works have used the MIMIC dataset, such as [94], [95] and [96].

In this work, it focus on the clinical data, the LABEVENTS and LABITEMS tables. The Labevents table contains data of each patient’s ICU stay, as presented in Table 4; and Table 5 contains descriptions of the lab events. Considering the medical labs were conducted. Therefore, the relationship between medical lab tests and patient deterioration are investigated. Thus, it could identify which medical tests have a major effect on clinical decision making. For example, the following information is about a patient who was staying at the ICU and was given a medical test. The following information was recorded at that time:

• Subject_ID:	2
• Hadm_ID:	25967
• IcuStay_ID:	3
• ItemID:	50468
• Charttime:	6/15/2806 21:48
• Value:	0.1
• ValueNum:	0.1
• Flag:	abnormal
• ValueUOM:	K/uL

Table 4: Lab events table description

Name	**Type**	**Null**	**Comment**
SUBJECT_ID	NUMBER(7)	N	Foreign key, referring to a unique patient identifier
HADM_ID	NUMBER(7)	Y	Foreign key, referring to the hospital admission ID of the patient
ICUSTAY_ID	NUMBER(7)	Y	ICU stay ID
ITEMID	NUMBER(7)	N	Foreign key, referring to an identifier for the laboratory test name
CHARTTIME	TIMESTAMP(6) WITH TIME ZONE	N	The date and time of the test
VALUE	VARCHAR2(100)	Y	The result value of the laboratory test
VALUENUM	NUMBER(38)	Y	The numeric representation of the laboratory test if the result was numeric
FLAG	VARCHAR2(10)	Y	Flag or annotation on the lab result to compare the lab result with the previous or next result
VALUEUOM	VARCHAR2(10)	Y	The units of measurement for the lab result value

Table 5: Lab items table description

Name	**Type**	**Null**	**Comment**
ITEMID	NUMBER(7)	N	Table record unique identifier, the lab item ID
TEST_NAME	VARCHAR2(50)	N	The name of the lab test performed
FLUID	VARCHAR2(50)	N	The fluid on which the test was performed
CATEGORY	VARCHAR2(50)	N	Item category
LOINC_ CODE	VARCHAR2(7)	Y	LOINC code for lab item
LOINC_DESCRIPTION	VARCHAR2(100)	Y	LOINC description for lab item

3.1.1 Medical Lab Tests Average Dataset

The dataset was constructed by taking the average test result of each patient for each kind of test and make it one attribute. Thus, one patient would be represented as one instance having 700 attributes, one for each test. If a test was not done, then the value of that attribute would be 0. For example, the first patient record in the dataset would look like this:

P_ID	Avg1	Avg2	.....	Avg700	Dead/Alive
1	5.3	10		0	D

3.1.2 Total Number of Medical Lab Tests Dataset

The dataset was built by taking the total number of tests taken for each patient for each type of test and make it one attribute. Then, one patient would be represented as one instance having 700 attributes, one for each test. If a test was not done, then the value of that attribute would be 0. For example, the dataset would look like this:

P_ID	Count1	Count2	…	Count700	Dead/Alive
1	5	0		1	D

3.2 Experiments

In the experiment section it investigate the effect of feature selection in improving the prediction of patient deterioration in the ICUs. It consider the lab tests as features. Thus, choosing a subset of features would mean choosing the most essential lab tests to perform. If the number of tests can be reduced by identifying the most critical tests, then it would also identify the redundant tests.

3.2.1 Experiment 1: Building a Baseline of the Medical Lab Tests Average

Experiment Goal

The goal of this experiment is to investigate the effect of lab testing on predicting patient deterioration. Usually, medical professionals compare the result of the lab test with a reference range [97]. If the value is not within this range, the patient may face fatal consequences. Thus, the patient is kept under observation and the test is repeated again during a specific period. In this experiment, it investigate the average value of the same repeated test and, more precisely, how the average value of lab results could assist medical professionals in evaluating patient status.

Since it dealt with real cases, the only way to assess the quality and characteristics of a data mining model was through the final status of the patient, i.e. whether the patient survived or not. Thus, the evaluation criterion was how accurately this proposed approach could predict whether the patient died or not.

Building the Dataset

The dataset was constructed by taking the average test result of each patient for each kind of test and make it one attribute. Thus, one patient would be represented as one instance having 700 attributes, one for each test. If a test was not done, then the value of that attribute would be 0. For example, the first patient record in the dataset would look like this:

P_ID	Avg1	Avg2	.....	Avg700	Dead/Alive
1	5.3	10		0	D

Pre-processing

After building the dataset, some values could not be reported because they were in text format. It used default values for these types of data. The total number of attributes was 619 with 2900 instances.

Base Learners

In this experiment it is used five classification algorithms to construct the model, namely *NaiveBayes*, *SMO*, *ZeroR*, *J48* and *RandomForest*.

Table 6: Experiment 1 confusion matrix results

Algorithm	**Learning Machine**	**Detailed Accuracy**			
		Accuracy	**Precision**	**Recall**	**F-Measure**
Bayes	NaïveBayes	42.96%	0.672	0.430	0.404
Functions	SMO	76.86 %	0.759	0.769	0.762
Rule	ZeroR	70.24 %	0.493	0.702	0.580
Tree	J48	75.27%	0.749	0.753	0.751
Tree	RandomForest	77.58 %	0.765	0.776	0.762

Evaluation

For a performance measurement, a 10-fold cross-validation of the dataset, and the confusion matrix was obtained to estimate four measures: accuracy, sensitivity, specificity and F-measure. As a result, *RandomForest* had the highest accuracy of 77.58%, followed by SMO with 76.86%, J48 with 75.27%, *ZeroR* with 70.24% and *NaïveBayes* with 42.96%, as shown in Table 6, Figure 10 and Figure 11. *RandomForest* and SMO have the same F-measures. The reason for the best performance by

RandomForest is that it works relatively well when used with high-dimensional data with a redundant/noisy set of features [88].

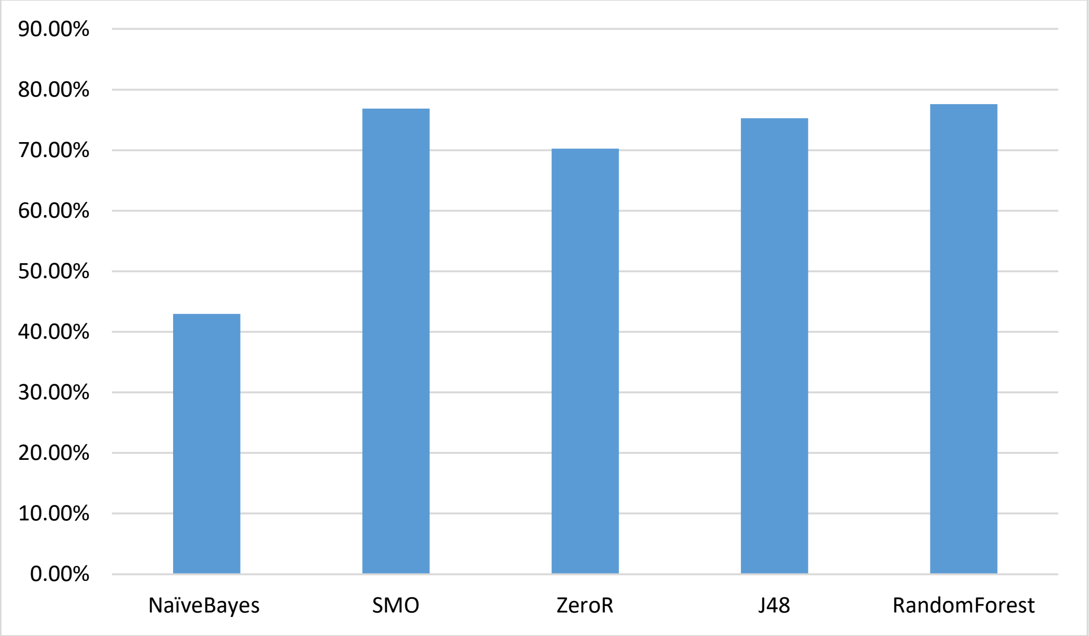

Figure 10: Experiment 1 accuracy result

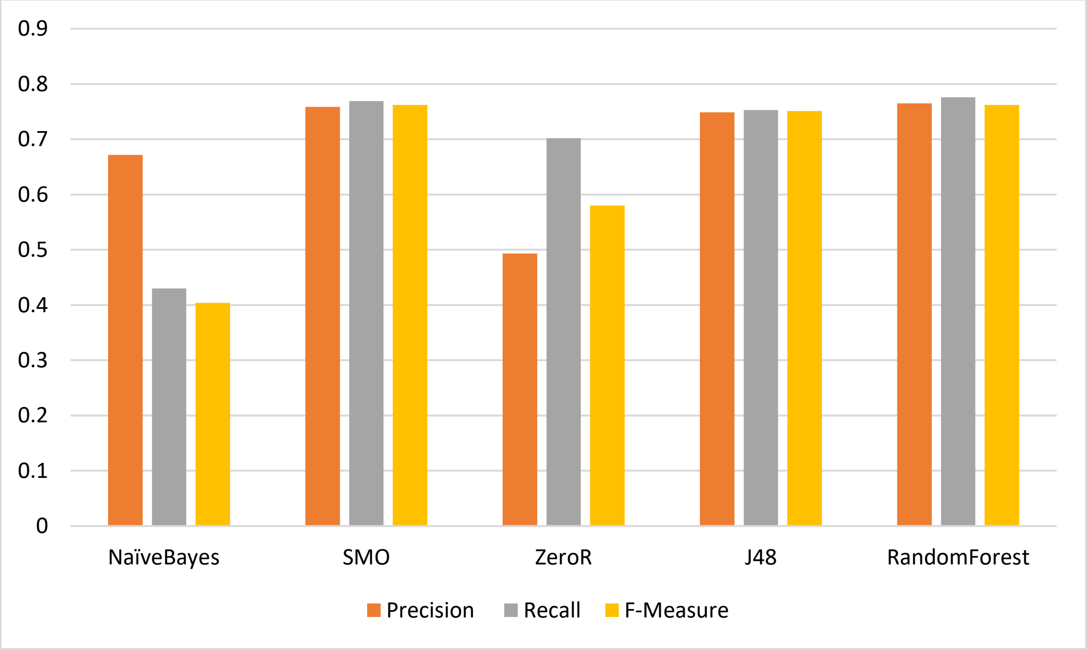

Figure 11: Experiment 1 detailed accuracy result

3.2.2 Experiment 2: Average Medical Lab Tests Discriminative Attributes

Experiment Goal

The goal of this experiment was to select the most discriminative attributes that can almost describe the model with a smaller number of attributes. This experiment is investigating the dependence between the average medical lab tests data and patient deterioration. Therefore, it would have a better understanding of patient deterioration problem.

Building the Dataset

This experiment used the same dataset in experiment 1 at Section 3.2.1.

Pre-processing

At this stage, feature selection is used to select the most discriminative attributes. For feature selection, it used *weka.attributeSelection.CfsSubsetEval* from WEKA [98].

- Attribute Subset Evaluator: CfsSubsetEval
- Search Method: BestFirst.
- Evaluation mode: evaluate all training data

Base Learner

Applying *CfsSubsetEval* reduced the attributes to 26 selected attributes. Now the goal was to compare the reduced dataset with the baseline experiment result. It used the same five classification algorithms to construct the model, namely *NaiveBayes, SMO, ZeroR, J48 and RandomForest.* Please refer to Table 7.

Table 7: Experiment 2 confusion matrix result

Algorithm	**Learning Machine**	**Detailed Accuracy**			
		Accuracy	**Precision**	**Recall**	**F-Measure**
Bayes	NaïveBayes	56.24 %	0.774	0.562	0.564
Functions	SMO	74.82 %	0.732	0.748	0.717
Rule	ZeroR	70.24 %	0.493	0.702	0.580
Tree	J48	76.75 %	0.765	0.768	0.766
Tree	RandomForest	79.75 %	0.790	0.798	0.789

Evaluation

Comparing the accuracy results from this experiment and the first experiment was reported in Table 8. As a result, the *NavieBayes* accuracy had the most significant increase, where it increased by 13 %. J48 and *RandomForest* had improved the result slightly. However, *SMO* and *ZeroR* did not have any enhancement at their accuracy result. Please refer to Table 8 and Figure 12.

Table 8: Accuracy comparison between Experiment 1 & Experiment 2

Algorithm	**Learning Machine**	**Accuracy of the original average dataset**	**Accuracy of the reduced average dataset**	**Change**
Bayes	NaïveBayes	42.96%	56.24 %	13.28%
Functions	SMO	76.86 %	74.82 %	-2.04%
Rule	ZeroR	70.24 %	70.24 %	0.00%
Tree	J48	75.27%	76.75 %	1.48%
Tree	RandomForest	77.58 %	79.75 %	2.17%

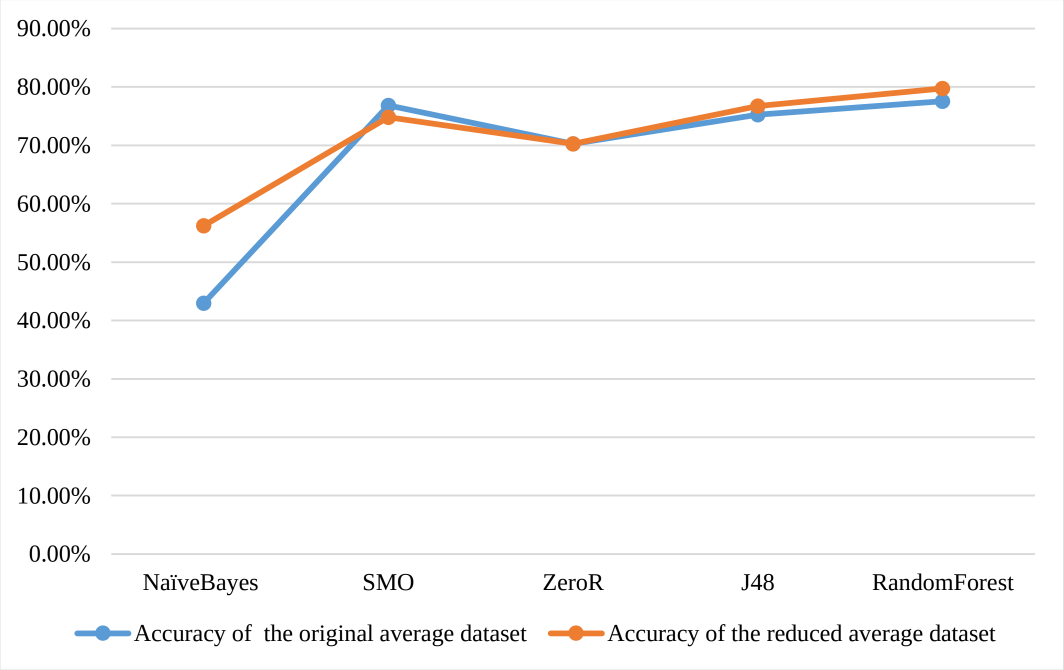

Figure 12: Accuracy comparison between Experiment 1 & Experiment 2

3.2.3 Experiment 3: Average Medical Lab Tests Feature Selection

Experiment Goal

The goal of this experiment was to study the relationship between feature selection and classification accuracy. Feature selection is one of the dimensionality reduction techniques for reducing the attribute space of a feature set. More precisely, it determines how many features should be enough to give reasonable accuracy.

Building the Dataset

This experiment used the same dataset as experiment 1 Section 3.2.1.

Pre-processing

This experiment built ten datasets depending on the number of selected features. It start with the first dataset, which contained only 10% of the total attributes. Then each time, it increased the total feature selections by 10%. For example, dataset 1 contains 10% of the total attributes, while dataset 2 contains 20% of the total attributes, dataset 3 contains 30% of the total attributes and so on till dataset 10 contains all 100% of the total attributes.

For feature selection, it *use supervised.attribute*. *InfoGainAttributeEval* from WEKA. This filter is a wrapper for the Weka class that computes the information gain on a class [98].

- Attribute Subset Evaluator: InfoGainAttributeEval
- Search Method: Ranker.
- Evaluation mode: evaluate all training data

Base Learner

After generating all the reduced datasets, it used the J48 algorithm to construct a model.

Table 9: Experiment 3 feature selection result

% of Features Selected	# of Features Selected	J48 Detailed Accuracy		
		Accuracy	**Number of leaves**	**Size of the Tree**
10%	62	75.10%	200	399
20%	124	73.59%	201	401
30%	186	75.10%	185	369
40%	248	74.93%	179	357
50%	310	75.17%	189	377
60%	371	74.79%	187	373
70%	433	75.00%	189	377
80%	495	75.31%	184	367
90%	557	74.97%	183	365
100%	619	74.86%	184	367

Evaluation

For each reduced dataset, 10-fold cross-validation for evaluating the accuracy is applied. Table 9 shows the results in numbers, and Figure 13 shows them as a chart. The results indicate that taking only the most related 10% of the total features can give 75.10% accuracy, which is comparable to the accuracy of the full feature set. This indicates that not all the features are required to get the highest accuracy. However, there are some fluctuations, such as at 20%, the accuracy drops a little. It is concluded that selecting 50 to 80% of the attributes' selection should give moderately satisfying accuracy.

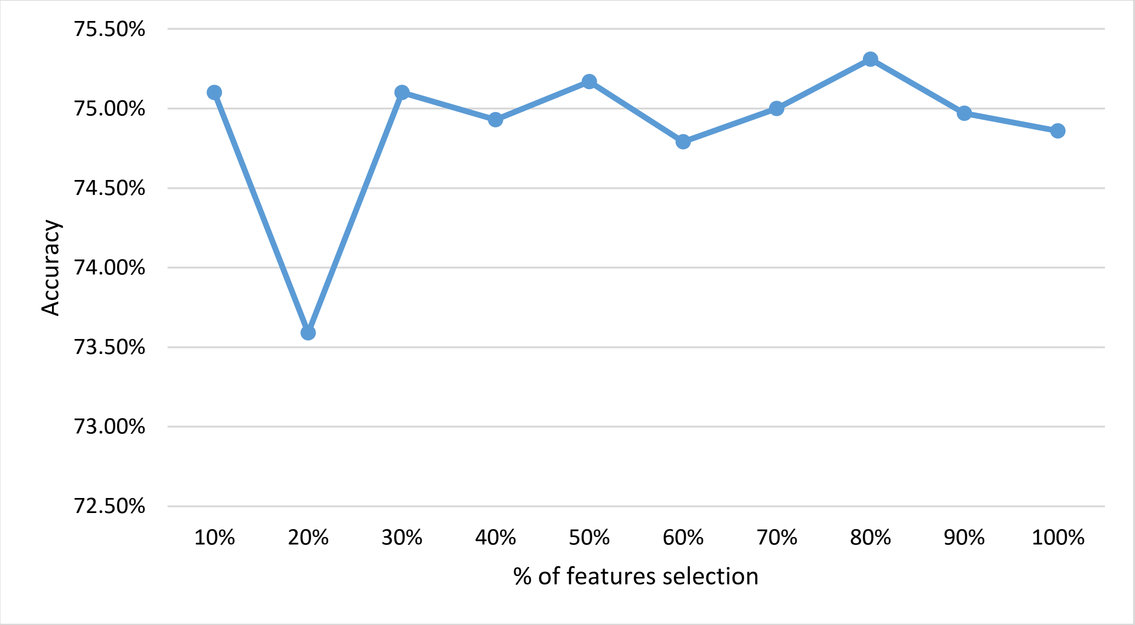

Figure 13: Average datasets accuracy

3.2.4 Experiment 4: Building a Baseline for the Total Number of Medical Lab Tests

Experiment Goal

The goal of this experiment was to investigate the effect of the total number of lab tests conducted on predicting patient deterioration. Usually, medical professionals keep requesting the same medical test over a brief period to compare the result with a reference range [97]. If the value is not within the range, the patient may be in danger, so the test is repeated again and again. The goal was to predict at what total number a medical professional should start immediate action and, more precisely, how the total number of medical lab tests could assist the medical professional in evaluating the patient’s status.

Building the Dataset

The dataset was built by taking the total number of tests taken for each patient for each type of test and make it one attribute. Then, one patient would be represented as one instance having 700 attributes, one for each test. If a test was not done, then the value of that attribute would be 0. For example, the dataset would look like this:

P_ID	Count1	Count2	…	Count700	Dead/Alive
1	5	0		1	D

Pre-processing

The dataset was randomized first, then two datasets were generated, *Count_Training_Validation_Dataset* and *Count_testing_Dataset*. This step was repeated ten times because it used randomization to distribute the instances between the two datasets.

Base Learners

Five learning algorithms were used to build the model, namely *NaiveBayes*, *SMO*, *ZeroR*, *J48* and *RandomForest*.

Table 10: Experiment 4 confusion matrix results

Algorithm	**Learning Machine**	**Detailed Accuracy**			
		Accuracy	**Precision**	**Recall**	**F-Measure**
Bayes	NaïveBayes	73.48%	0.716	0.735	0.711
Funtions	SMO	74.85%	0.737	0.749	0.716
Rule	ZeroR	69.72%	0.486	0.697	0.573
Tree	J48	72.44%	0.722	0.724	0.723
Tree	RandomForest	75.30%	0.739	0.753	0.736

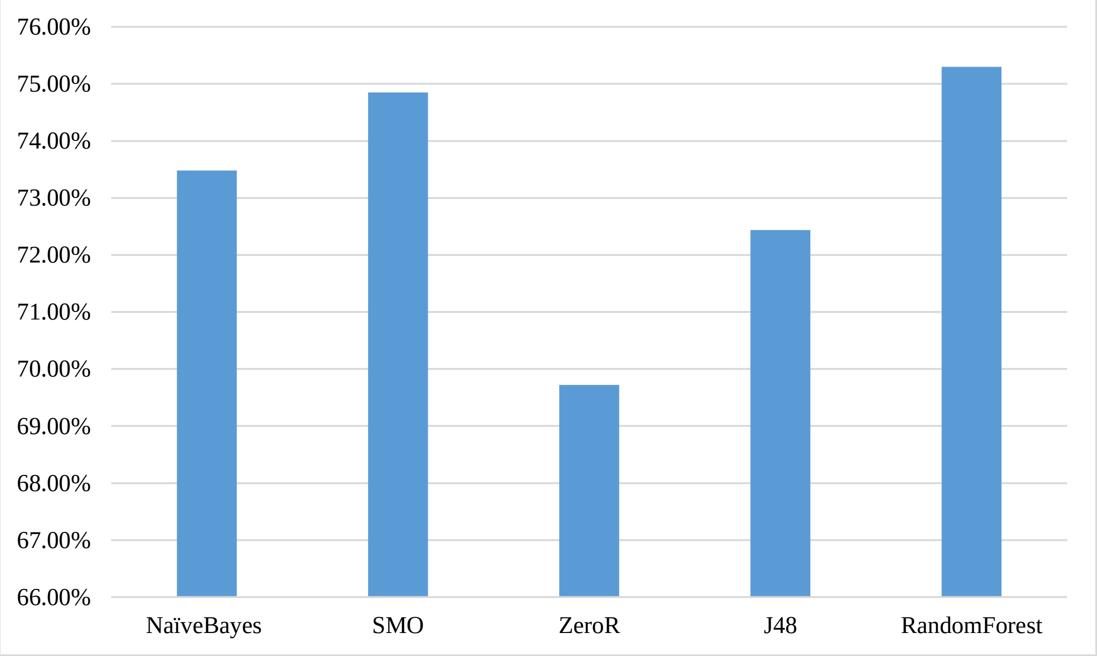

Figure 14: Experiment 4 accuracy result

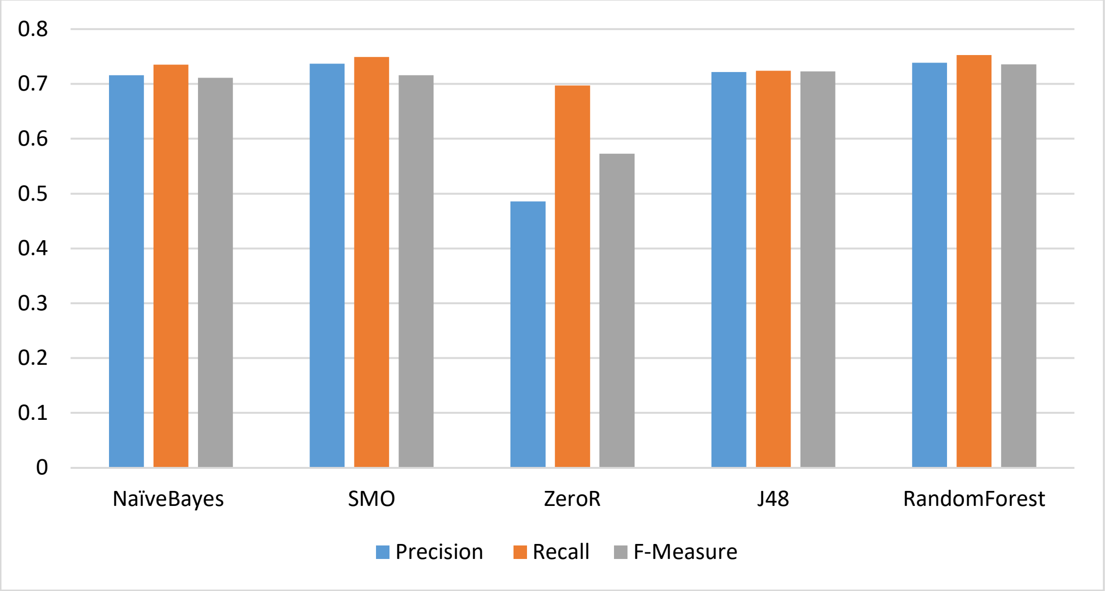

Figure 15: Experiment 4 detailed accuracy result

<u>*Evaluation*</u>

The training data were first used to build the model and then evaluated using a percentage split via test data. For a performance measurement, the confusion matrix was obtained to estimate four measures: accuracy, sensitivity, specificity and F-measure. Table 10 shows that *SMO* and *RandomForest* have almost equal levels of accuracy, around 75%. Even after testing the model with the test datasets, *SMO* and *RandomForest* still have the highest accuracy among the other techniques. The reason for this higher accuracy is that the amount of memory required for *SMO* is linear in the training set size, which allows *SMO* to handle extensive training sets [99].

3.2.5 Experiment 5: Total Number of Medical Lab Tests Discriminative Attributes

Experiment Goal

The goal of this experiment was to select the most discriminative attributes that can almost describe the model with less number of attributes. This experiment was aiming to get the most out of the total number of medical lab tests data, so it could have a better understanding to patient deterioration problem.

Building the Dataset

This experiment used the same dataset in experiment 4.

Pre-processing

In this stage, feature selection is used to select the most discriminative attributes. For feature selection, it used *weka.attributeSelection.CfsSubsetEval* from WEKA [98].

- Attribute Subset Evaluator: CfsSubsetEval
- Search Method: BestFirst.
- Evaluation mode: evaluate all training data

Base Learner

Applying *CfsSubsetEval* reduced the attributes to 26 selected attributes. Now the goal was to compare the reduced dataset with the baseline experiment result. It used the same five classification algorithms to construct the model, namely *NaiveBayes, SMO, ZeroR, J48 and RandomForest*.

Table 11: Experiment 5 confusion matrix results

Algorithm	**Learning Machine**	**Detailed Accuracy**			
		Accuracy	**Precision**	**Recall**	**F-Measure**
Bayes	NaïveBayes	73.17 %	0.709	0.732	0.702
Functions	SMO	73.68 %	0.726	0.737	0.684
Rule	ZeroR	70.24 %	0.493	0.702	0.580
Tree	J48	73.82 %	0.726	0.738	0.730
Tree	RandomForest	74.65 %	0.731	0.747	0.733

Evaluation

Comparing the accuracy results from this experiment and the fourth experiment was reported in Table 11 and Table 12. As a result, there was no enhancement in general. Only J48 1.38%.

Table 12: Accuracy comparison between Experiment 4 & Experiment 5

Algorithm	**Learning Machine**	**Accuracy of the original total number of tests dataset**	**Accuracy of the reduced total number of tests dataset**	**Change**
Bayes	NaïveBayes	73.48%	73.17 %	-0.31%
Functions	SMO	74.85%	73.68 %	-1.17%
Rule	ZeroR	69.72%	70.24 %	0.52%
Tree	J48	72.44%	73.82 %	1.38%
Tree	RandomForest	75.30%	74.65 %	-0.65%

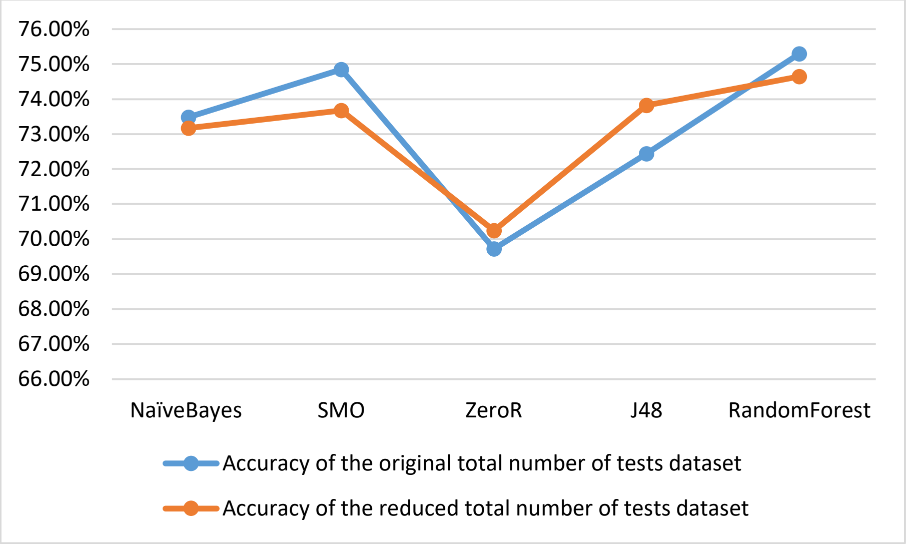

Figure 16: Accuracy comparison between Experiment 4 & Experiment 5

3.2.6 Experiment 6: Feature Selection for Total Number of Medical Lab Tests

Experiment Goal

The goal of this experiment was to study the relationship between feature selection and classification accuracy. In other words, how many features should be enough to give reasonable accuracy?

Building the Dataset

This experiment used the count dataset.

Pre-processing

This pre-processing step built ten datasets depending on the number of selected features. The first dataset contained only 10% of the total attributes. Then it increased the total feature selections by 10% with each new dataset. For example, dataset 1 contained 10% of the total attributes, dataset 2 contained 20% of the total attributes, dataset 3 contained 30% of the total attributes and so on till dataset 10 contained all 100% of the total attributes.

For feature selection, it used *supervised.attribute*. *InfoGainAttributeEval* from WEKA. This filter is a wrapper for the Weka class that computes the information gain on a class [98].

- Attribute Subset Evaluator: InfoGainAttributeEval
- Search Method: Ranker.
- Evaluation mode: evaluate on all training data

Base Learner

After generating all reduced datasets, the J48 algorithm is used as a base learner.

Table 13: Experiment 4 results

% of Features Selection	**# of Features Selection**	**Detailed Accuracy**		
		Accuracy	**Number of leaves**	**Size of the Tree**
10%	62	71.45%	237	473
20%	124	73.90%	250	499
30%	186	73.55%	247	493
40%	248	72.79%	252	503
50%	310	73.41%	252	503
60%	371	73.66%	254	507
70%	433	74.24%	254	507
80%	495	74.10%	254	507
90%	557	74.14%	265	529
100%	619	73.59%	259	517

Evaluation

Each feature-reduced dataset went through a 10-fold cross-validation for evaluation. Figure 17 shows the accuracy of all count datasets. The detailed values are also reported in Table 13. From the results it is observed that selecting 60 to 70% of the attributes gives the highest accuracy. This also concludes that all features (i.e., lab tests) may not be necessary to attain a highly accurate prediction of patient deterioration.

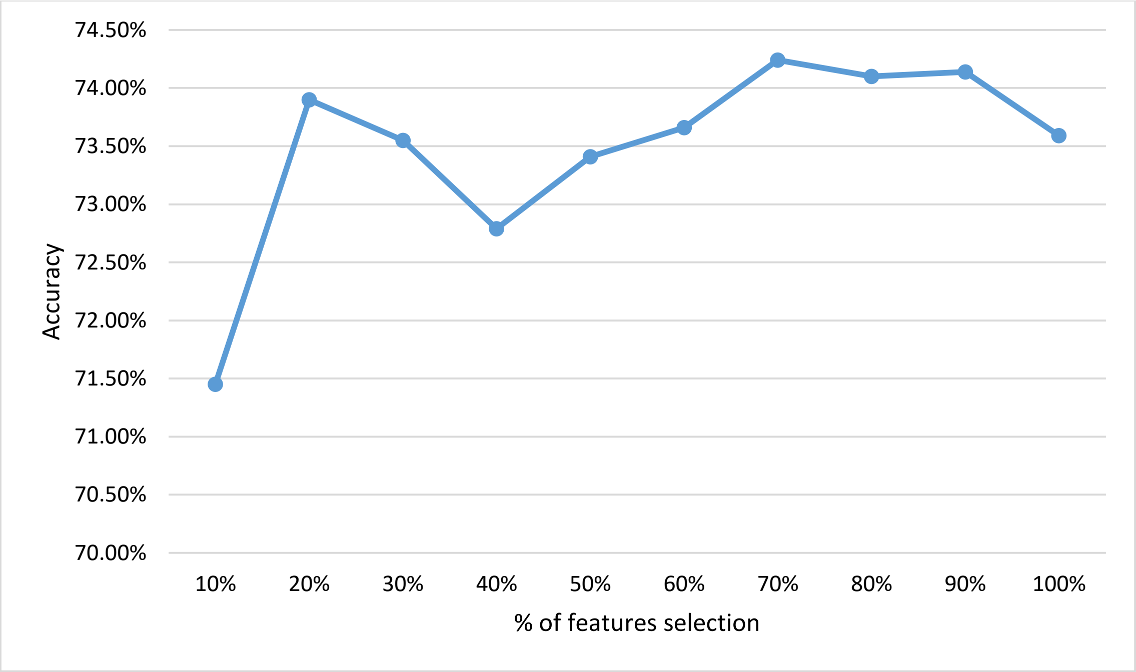

Figure 17: Count Dataset accuracy

3.3 Discussion

The previous experiments investigated the effect of feature selection in improving the prediction of patient deterioration in the ICUs. They considered the lab tests as features. Thus, choosing a subset of features would mean choosing the most important lab tests to perform. If the number of tests could be reduced by identifying the most important tests, then it would also identify the redundant tests. It should be noted that feature selections was made without any domain knowledge and without any intervention from medical experts. However, in the analysis it would like to emphasize the merit of feature selection in choosing the best tests, which could be further verified and confirmed by a medical expert.

First, it compare the selected features selected from the two datasets, namely the average dataset and the count dataset. Table 14 shows the ten best features chosen by the two approaches and highlights the standard lab tests between the two approaches (i.e. using the average of tests and count of tests). Table 15 shows more details about the common tests.

Table 14: Final results

	Average Dataset	**Count Dataset**
Best ranked 10 from the 10% of selected features	50177	50148
	50090	50112
	50060	50140
	50399	50399
	50386	50177
	50440	50439
	50408	50090
	50439	50440
	50112	50079
	50383	50068

Table 15: Medical lab test details

	Detailed Description				
	Test_Name	**Fluid**	**Category**	**LOINC _Code**	**LOINC_Desc**
50177	UREA N	BLOOD	CHEMISTRY	3094-0	Urea nitrogen [mass/volume] in serum or plasma
50090	CREAT	BLOOD	CHEMISTRY	2160-0	Creatinine [mass/volume] in serum or plasma
50399	INR(PT)	BLOOD	HEMATOLOGY	34714-6	INR in blood by coagulation assay
50440	PTT	BLOOD	HEMATOLOGY	3173-2	Activated partial thromboplastin time (aPTT) in blood by coagulation assay
50439	PT	BLOOD	HEMATOLOGY	5964-2	Prothrombin time (PT) in blood by coagulation assay
50112	GLUCOSE	BLOOD	CHEMISTRY	2345-7	Glucose [mass/volume] in serum or plasma

LOINC is an abbreviation for logical observation identifiers names and codes. LOINC is clinical terminology important for laboratory test orders and results [100]. ARUP Laboratories [101] is a national clinical and anatomic pathology reference laboratory and a worldwide leader in innovative laboratory research and development. Table 10 clarifies more about the medical lab tests as follows:

- UREAN (50177): This test is conducted using the patient’s blood. This test is recommended to screen for kidney dysfunction in patients with known risk factors (e.g. hypertension, diabetes, obesity, family history of kidney disease). The panel includes albumin, calcium, carbon dioxide, creatinine, chloride, glucose, phosphorous, potassium, sodium and BUN and a calculated anion gap value. Usually, the result is reported within 24 hours [101].

- CREAT (50090): This test is conducted using the patient’s blood. It is a screening test to evaluate kidney function [101].
- INR(PT) (50399): This test is conducted using the patient’s blood by coagulation assay [93].
- PTT (50440): This test is carried out to answer two main questions: does the patient have antiphospholipid syndrome (APLS), and does the patient have von Willebrand disease? If so, which type? It is carried out by mechanical clot detection [102].
- PT (50439): This test is conducted using the patient’s blood by coagulation assay [93].
- GLUCOSE (50112): This test is used to check glucose, which is a common medical analytic measured in blood samples. Eating or fasting prior to taking a blood sample has an effect on the result. Higher than usual glucose levels may be a sign of prediabetes or diabetes mellitus [102].

- The result of the top 10 selected features from the average dataset allows us to build a model using decision tree J48. This model would allow a medical professional to predict the status of a patient in the ICU as follows:

```
50440 <= 20.757143: 1 (772.0/22.0)
50440 > 20.757143
|  50177 <= 25.923077
|  |  50060 <= 0
|  |  |  50112 <= 138.333333
|  |  |  |  50383 <= 28.155556
|  |  |  |  |  50112 <= 110.470588
|  |  |  |  |  |  50399 <= 1.204545: 0 (5.0)
```

For example, if the lab test (name: PTT, ID 50440, LOINC: 3173-2) result value is <= 20.757143, then the probability is very high (772.0/22.0~ 97.2%) that the patient is going to die (class:1). This model has 78.6897% overall accuracy.

3.4 Finding and Further Research

The increasing amount of medical laboratory data represents a significant information resource that can provide a foundation for the improved understanding of patients' critical. Data mining supports this goal by providing a set of techniques designed to discover similarities and relationships between data elements in large data sets.

Reducing frequent laboratory testing and the potential care and financial implications are critical issues in the intensive care units. In this dissertation, it presented the proposed approach to reduce the observation time in the ICU by predicting patient deterioration in its early stages. In this work, it presented six experiments to investigate the effect of the average laboratory test value and the number of the total laboratory in predicting patient deterioration in the Intensive Care Unit. In this work, it considered laboratory tests as features. Choosing a subset of features would mean choosing the most essential lab tests to perform.

For future work, the authors are planning to carry out more experiments using bigger data. Big data analytics would bring potential benefits to support taking the right decision to enhance the efficiency, accuracy and timeliness of clinical decision making in the ICU. Besides that, this dissertation is planning to use streaming feature selection approaches in the future to study more this case.

In the next chapter, an overview of the proposed streaming feature selection approach is provided.

Chapter 4: Proposed Streaming Feature Grouping and Selection Approach

In this chapter, an overview of the proposed SFGS approach is provided to applying feature selection in a streaming manner.

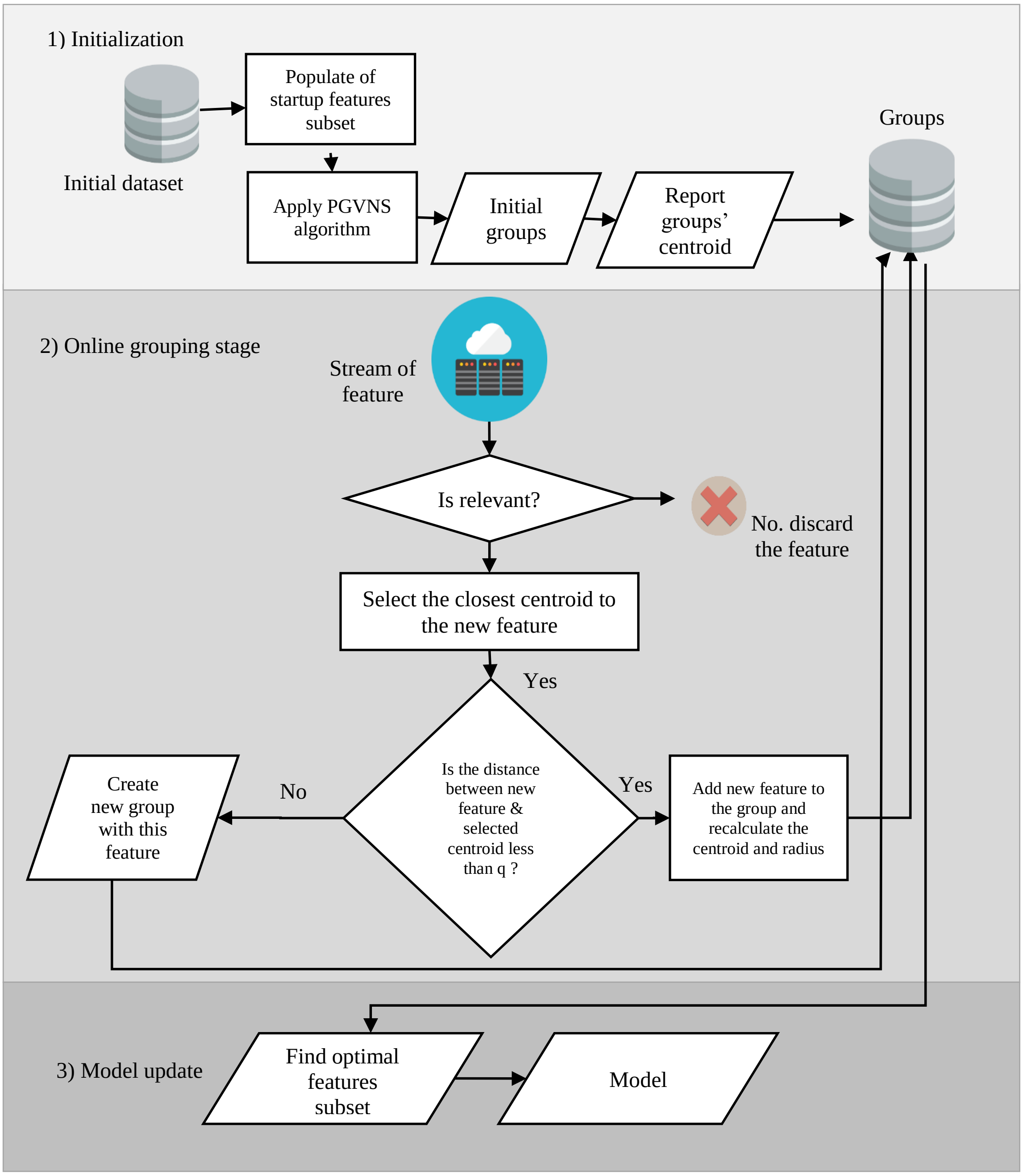
1) Initialization
Initial dataset
Populate of startup features subset
Apply PGVNS algorithm
Initial groups
Report groups' centroid
Groups
2) Online grouping stage
Stream of feature
Is relevant?
No. discard the feature
Select the closest centroid to the new feature
Yes
Is the distance between new feature & selected centroid less than q ?
No
Yes
Create new group with this feature
Add new feature to the group and recalculate the centroid and radius
3) Model update
Find optimal features subset
Model

Figure 18: SFGS high-level design consists of three stages (1) initialization, (2) online grouping for assigning new coming relevant feature, and (3) model update to recalculate the groups' centroids and final most representative subset selection.

The proposed SFGS algorithm consists of three main stages as illustrated in Figure 18, namely, initialization, online grouping, and model update.

1) Initialization stage: it start with an initial dataset. This dataset consists of part of the features in the stream. It apply the “Predominant Group-based Variable Neighborhood Search” (PGVNS) algorithm [26] on the initial feature set in order to partition the features into groups as specified in definition 6. Then, it report the centroid and radius of each resulted group and save them. Please refer to Figure 19.

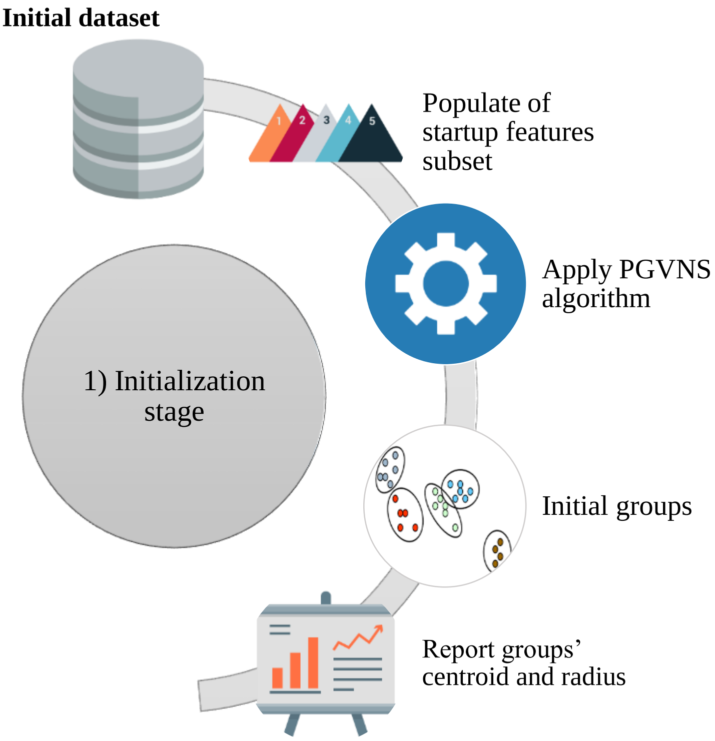

Figure 19: Initialization stage

2) Online grouping stage: because of the features stream problem, it assume that the stream of features arrives in batches (or subsets) of features. Accordingly, each new feature is assessed upon its arrival to determine whether to accept it or not. The assessment evaluates the worth of each feature f_i by measuring the symmetrical uncertainty with respect to the class as stated in definition 4.

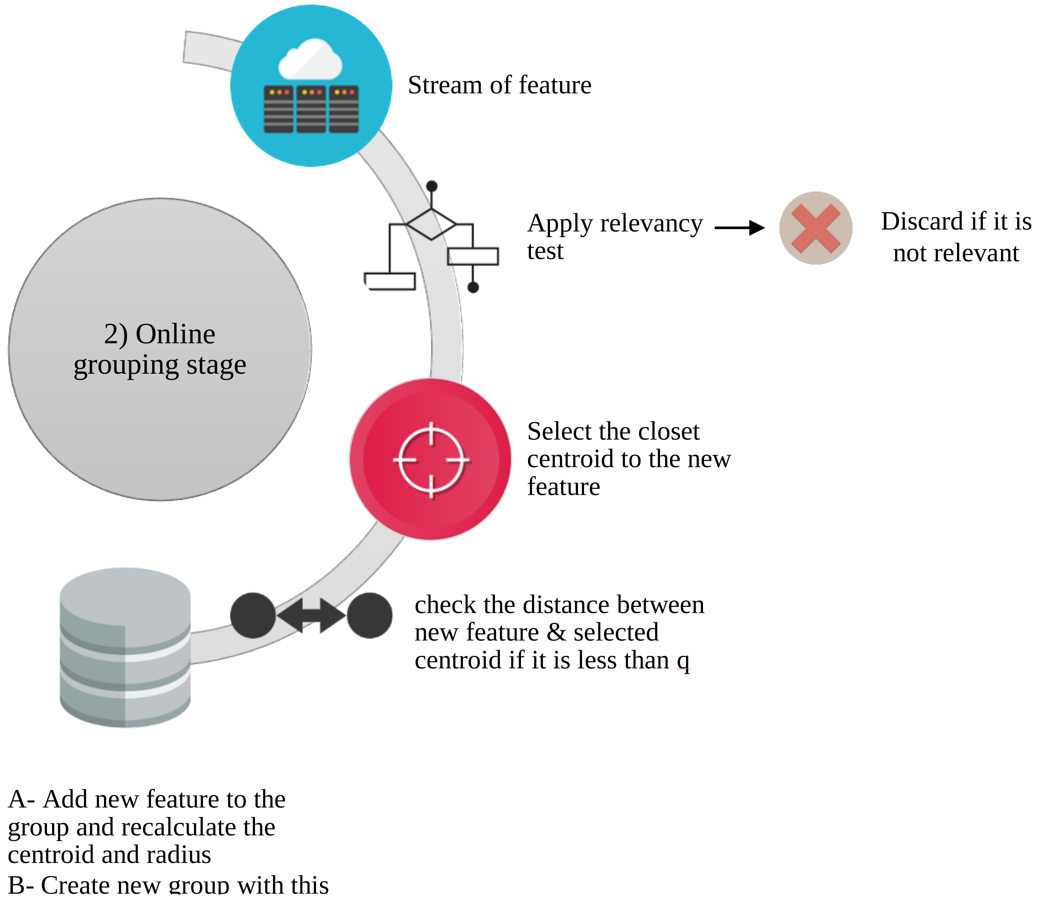
Stream of feature
Apply relevancy test
Discard if it is not relevant
2) Online grouping stage
Select the closet centroid to the new feature
check the distance between new feature & selected centroid if it is less than q
A- Add new feature to the group and recalculate the centroid and radius
B- Create new group with this

Figure 20: Online grouping stage

After accepting the relevant feature, the feature is evaluated if it is redundant for other existing features. This assessment will be achieved using feature grouping. Each group has a group midpoint called centroid (see definition 7), which is the most-representative feature of the group to which all other features are the most similar. This can be achieved by finding the similarity between all group members (features) to nominate a centroid as in definition 5. Besides that, radius is calculated according to definition 8. The new relevant feature will be either allocated to one of the available feature groups or it can formulate a new group, depending on the distance of the feature from the groups and the average radius of all groups. Please refer to Figure 20.

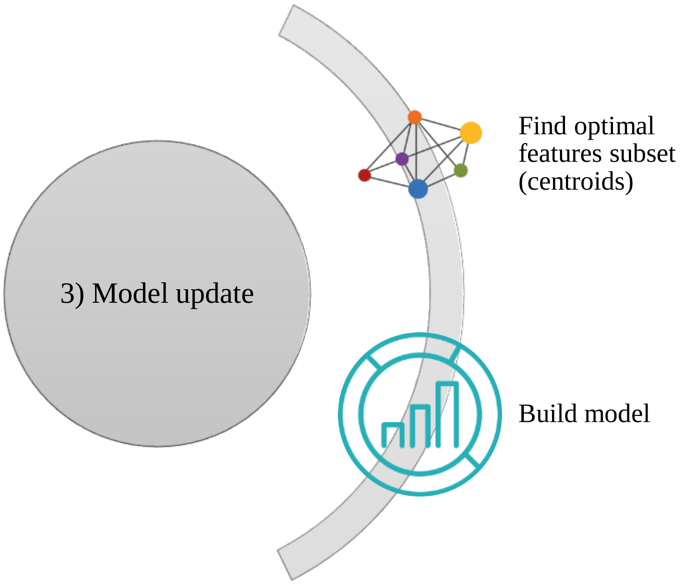

Figure 21: Model update

3) Model update: after handling a batch of features in the features stream, new centroids from each group will be computed. The groups will be updated each time there is a new features stream. The centroid of each group will be used as a feature for a learning model. Please refer to Figure 21.

- **Definition 1 (Feature f_i)**: A feature f_i *is the i-th feature received from the feature stream.*

- **Definition 2 (Initial *Feature Set, F*)**: The initial set of features F is a collection of features F $= \{ f_1, \ f_2, \dots , f_k\}$ *where k is size of the feature set.*

- **Definition 3 (*Feature stream,* *S*)**: is a set of features, where the full set is unknown at the beginning. The new features appear one by one over time, but the total sample size remains fixed. The proposed approach applies the "Predominant Group-based Variable Neighborhood Search" (PGVNS) algorithm [26] on the initial feature set in order to partition the features into groups. Accordingly, each new feature is assessed upon its arrival to determine whether to accept it. The proposed approach evaluates the worth of each feature f_i by measuring the symmetrical uncertainty with respect to the class as stated in definition 4.

- **Definition 4 (Feature *Relevance criteria*)**: Relevance criteria is the measures to evaluate how a single feature relevant to the selected class, where this feature is essential for the final most representative subset. In this test, Gain ratio is applied which is a variant of the information gain that reduces its bias.

$$Gain\, ratio(f_i) = \frac{gain(f_i)}{intrinsic\ info(f_i)} \qquad (1)$$

- **Definition 5 (Distance between two features, *Dist* (f_a, f_b))**: The function Dist(f_a, f_b) denotes the distance between two features f_a and f_b measured using symmetrical uncertainty (SU) which is one of normalized form of Mutual Information (MI).

$$dist(f_a, f_b) = SU(f_a, f_b) = \sum_{f_a, f_b} p\ (f_a, f_b)\ log \frac{p(f_a, f_b)}{p(f_a)p(f_b)} \qquad (2)$$

$where\ p\ (f_a, f_b)\ is\ the\ joint\ probability\ distribution\ betwen\ f_a\ and\ f_b$

$a = 1\ to\ n, b = 1\ to\ n$ and n = total number of relevant features

After accepting the relevant feature, the proposed approach determines areas in which this feature is redundant for other existing features. This assessment will be achieved using feature grouping. Each group has a group midpoint called centroid (see definition 7), which is the most-representative feature of the group to which all other features are the most similar.

- **Definition 6 (Feature *Group* G_i)**: A feature group G_i *is a set of features that are more similar to each other than to those in other groups. The similarity is measured by Definition 5.*

$G_i = \{f_{i_1}, \dots, f_{i_n}\}\ where\ \ f_{i_j}$is j^{th}feature of group G_i

- **Definition 7 (*Group's Centroid*)**: *Each feature group has a group midpoint called centroid, which is the most representative feature, which is in effect the group's medoid based on the distance metric defined in definition 5:*

$$Centroid\ (G_i) = f_{m_i}, \text{such that}\ m = \underset{j=1\ to\ n}{Min\ Index} \sum_{\substack{k=1 \\ k\neq j}}^{n} dist\ (f_{i_j}, f_{i_k}) \quad (3)$$

This can be achieved by finding the similarity between all group members (features) to nominate a centroid as in definition 5.

- **Definition 8 (*Group's Radius* R_i)**: The radius R_i of Group G_i is *the distance between the group's centroid and the farthest member in that group.*

$$Radius\ (G_i) = Max_{j=1\ to\ n}\ dist(f_{i_j}, Centroid(G_i)) \qquad (4)$$

- **Definition 9 (Average Radius** (AvgRad)**)**: is the sum of all the groups radius divided by the total number.

$$\sum_{i=1}^{k} \frac{Radius(G_i)}{k} \qquad (5)$$

- **Definition 10** (**Distance** (f_j, G_i)):is the distance between new relevant feature and a group's centroid. $distance(f_j, G_i)$

- **Definition 11 (q):** is a user-defined parameter used to control the distance threshold for either creating a new cluster or placing new feature in existing clusters. It compares the average radius at Definition 9 with the nearest group from Definition 10 as follow: $\mathbf{AvgRad} * \mathbf{q} > \mathbf{distance}(\boldsymbol{f_j}, \boldsymbol{G_i})$.

A q is a constant number defined by the user. $G_i$is the nearest of f_j ; and f_j is used to create a new group. Otherwise f_j is included in the G_i.

4.1 SFGS Algorithm

Algorithm SFGS

```
1:  $\boldsymbol{F}$ ← initial set of features
2:  $\boldsymbol{G}$ ← $Initial\_Grouping$ ($\boldsymbol{F}$) // using PGVNS [26]
    /*      where $G = \{G_1, \ldots, G_k\}$ where k is the total number of groups
                 $G_i = \{f_{i_1}, \ldots, f_{i_n}\}$ $where$ $f_{i_j}$ is $j^{th}$ feature of group $G_i$
                 $Radius\ (G_i)$ ← apply equation (4) to get Radius of $G_i$ */
3:  avgRadius ← calculate_Avg_Radius ($G$) //apply equation (5)
4:  While (true) //continue until stream has no new features
5:      $f_j$ ← next feature in the stream
6:      $v$ ← $gainRatio(f_j)$  //apply equation (1)
7:      if ($v \le t$ ) break  //not relevant
8:      else // relevant
9:          for i=1 to k
10:             $d_i$ ← Distance($f_j$, $G_i$) //definition (11)
11:             $d_m$ ← $min$ ($d_1$, …, $d_k$)
12:             if ($d_m$ < q * avgRadius) // apply definition (12)
13:                 $G_m$ ← $G_m \cup f_j$ //put into this group
14:                 Centroid($G_m$) ← calculate_Centroid($G_m$)  // apply equation (3)
15:                 Radius($G_m$) ← calculate radius($G_m$) // apply equation (4)
16:             else // create a new group
17:                 $G_{k+1}$ ← $\{f_j\}$
18:             end if
19:         end for
20:         avgRadius ← calculate_Avg_Radius($G$) // recalculate average radius using equation (4)
21:     end if
22: end while
```

The algorithm shows a detailed step for the proposed SFGS algorithm. It assume that have an initial set of features ***F*** in Step 1. In Step 2 it is using a variant of PGVNS [26] algorithm to partition the features into groups and calculate the centroid and radius of each resulted group and save them. From Step 4 to Step 22 the online selection and grouping is processed. Each feature is checked first if it is relevant to the class or not as Step 6 using gain ratio which is a non-symmetrical measure that is used to overcome the limitation of the Information Gain (IG), where its selection for the informative feature is not affected by the large values of that feature. Thus, the resulted feature will be either relevant so it will go to the next step, or it will be discarded. In Step 10 and Step 11, the algorithm will find the distance using mutual information (MI) between the new candidate feature and existing groups' centroids. The new candidate feature will be assigned to the closest centroid depending on the distance between the new feature and centroid. This is done as follows: the algorithm will compare the AvgRadius * the value of q and the distance between new feature & selected centroid as according to definition 11. The proposed approach used a value close to 1.5 assuming the normal distribution of the radii.

Then, it will keep the new feature in the corresponding group (Step 13 to Step 15). Otherwise, it will create a new group and assign this feature as centroid (Step 17). The average radius of the groups will also be updated as Step 20. All incoming features will follow the Steps from Step 5 to Step 22.

Figure 22 represents an illustrative example of adding a new candidate-relevant feature to the existing groups. Note that the centroid of group 1 is the closest one. Thus, it will be assigned to Group 1, and the new centroid of the group will be allocated. In contrast, Figure 23 represents the other cases, in which the new candidate relevant feature will be in a new group by itself.

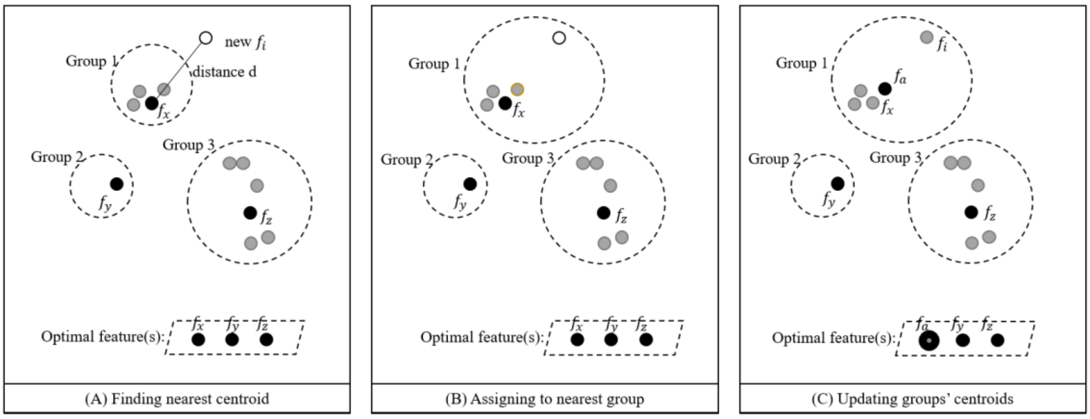

Figure 22: Shows an illustration scenario of adding a new candidate-relevant feature. Since the $\boldsymbol{dist(f_i, G_1) < q * AvgRad}$, the $\boldsymbol{f_i}$ will be assigned to Group 1 and the group's centroid will be redefined. Correspondingly, the most representative feature will be updated too.

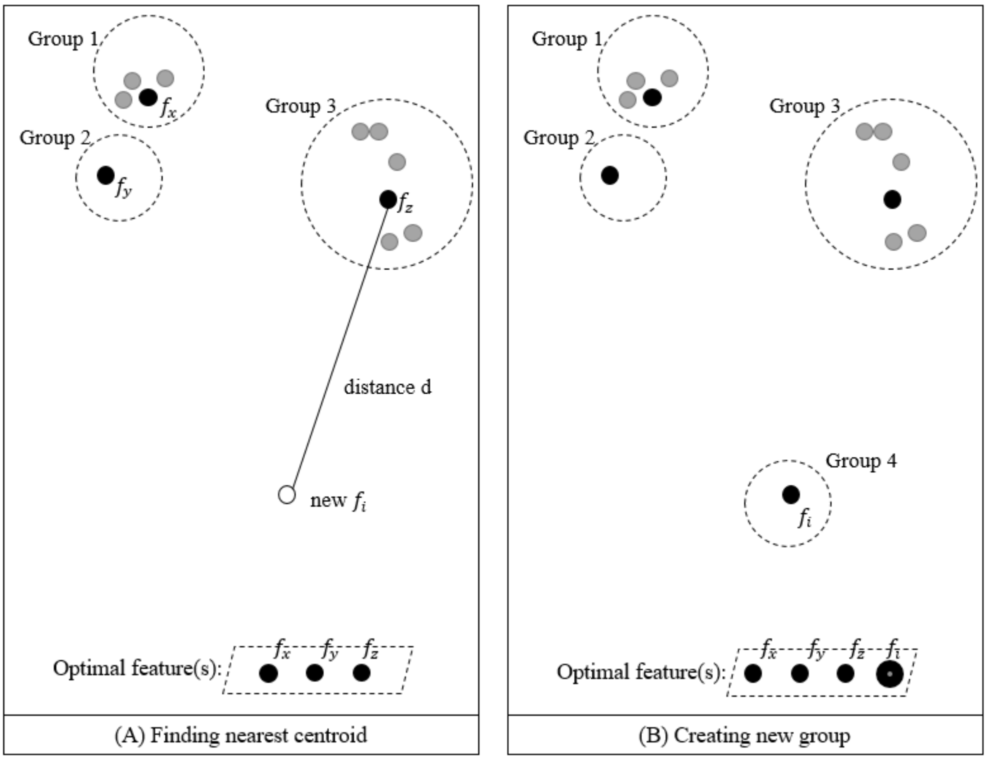

Figure 23: Presents the other case, in which the new candidate-relevant feature is in a new group by itself. Since the $\boldsymbol{dist(f_i, G_1) > q * AvgRad}$, the $\boldsymbol{f_i}$ will be assigned to a new group by itself and it will also be the new centroid. Correspondingly, the most representative feature will be updated too.

4.2 Analysis of Effect of q

In the proposed approach, it examine the impact of the q on generating the grouping. This analysis examines the trade-off between number of groups and the quality of the feature groups.

There are two extreme cases of the q: zero and infinity. When q=0, no feature is included in any existing groups because the condition $\boldsymbol{dist(f_i, G_1) < q * AvgRad}$ will be always false. Therefore, each feature will be in a singleton group, resulting in a grouping that is essentially the same as no grouping. However, this extreme case is unacceptable because it offers no feature reduction. The second extreme case is q= infinity, and in this case all features will be placed in the same group as the condition $\boldsymbol{dist(f_i, G_1) < q * AvgRad}$ will be always satisfied. In this case the centroid doesn't represent most of the features, which will ultimately result in a very poor classification model. Therefore, the SFGS choose a value of q that gives us the best tradeoff between number of groups and the quality of the groups, such that the total number of groups is less than the number of features and each group centroid represents the group members well.

It is understood that when the q increases, the size of the group's radius would be increased too. Therefore, the group quality is decreased, because the group's centroid would be less representative of that group. Consequently, if the q value increased, the average radius is gradually increased too. Besides, when the average cluster size increases, the total number of clusters is reduced. On the contrary, if the q value decreased, the average radius is gradually decreased increasing the total number of clusters. The following lines represents an assumption in mathematical representation for the function q.

Thus, if the $f(q) = \begin{cases} N, if\ q = 0 \\ 1, if\ q = \infty \\ 1 \leq r \leq N, if\ 0 < q < \infty \end{cases}$

N represents the number of features.

r represents the number of clusters.

$f(q)$ represents a monotonically decreasing function.

For example: if $q1 \leq q2$

Then, $f(q_1) \geq f(q_2)$

$r_1 \geq r_2$

Figure 24 below illustrates this observation. It empirically observed the best classification results when q value to would be between 1.5 to 2 (see Section 5.1.5).

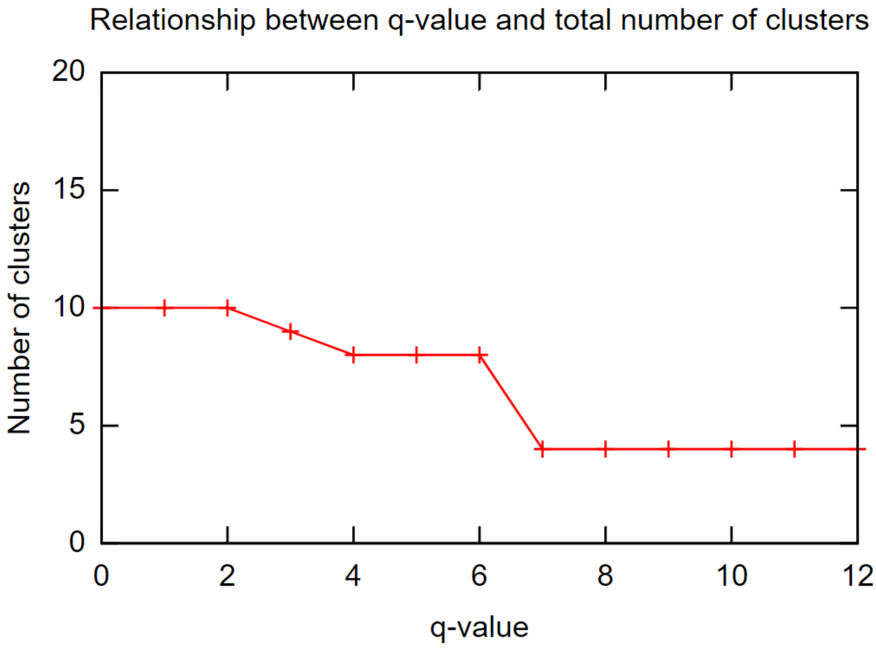

Figure 24: The effect of q on generating the most representative subset. You can notice that when the q increase, the total number of the generated groups is decreased until they reach a fixed number of groups.

4.3 Runtime Complexity

Runtime complexity uses the big-O notation to evaluate the efficiency of the proposed SFGS. Following from Section 4.1, the runtime of the SFGS algorithm comes across the next time complexities:

Let $n = |F|$ number of features.

Line 2: initializing groups would perform $O(k)$ operations, where k = number of groups.

Line 3: calculating average radius would perform only once, which is $O(\mathrm{k})$.

Line 6: calculating gain ratio execute $O(1)$.

Line 10: the for loop would be executed $O(k)$ times, and since step 15 & 16 takes $O(n)$ times, total execution time of the for loop is $O(nk)$

Line 13: the if statement would be executed $O(n)$

Line 20: calculating average radius would execute $O(k)$

Therefore, the complexity for each iteration is $O(nk)$, which is a linear time complexity that increases with the number of incoming features. Thus, if the total iteration is L, the overall complexity will be $O(Lnk)$.

In the next chapter, the SFGS experiment setup is described.

Chapter 5: Experimentation and Evaluation

In this chapter, the SFGS experiment is described and evaluated. The experiment setup begins with the benchmark datasets obtained from the UCI and provides each one's detailed properties. Then, it explore the learning algorithms used to evaluate classification performance. Furthermore, it illustrate the three state-of-art competing approaches to compare them with the proposed SFGS performance. It also presents the hardware and software environments. Last, it discusses the parameter setup. The experimental work contrasts with the SFGS results regarding the three competing algorithms: PGVNS, Fast-OSFS and Alpha-investing. Also it discusses the running-time performance and estimate the parameters' sensitivity in the experimental results.

5.1 Datasets

In this experiment work, it select four datasets to evaluate the performance of the proposed approach. These datasets are at different sizes and can be used for benchmarking deep learning algorithms. All datasets are obtained from Open Machine Learning [103] and UCI Machine Learning Repository [75] as follow, please refer also to Table 16:

1) ARCENE dataset [103][75] consists of mass-spectrometric data, which is used to distinguish cancer versus normal patterns. This is a two-class classification problem with continuous input variables. ARCENE's original owners are the National Cancer Institute (NCI) and the Eastern Virginia Medical School (EVMS). This dataset is one of five datasets of the NIPS 2003 feature selection challenge.

Table 16: reports more detailed information about the datasets [103][75]

Property	Arcene	Dorothea	Hiva agnostic	Madelon
Number of instances (rows) of the dataset.	200	1150	4229	2600
Number of attributes (columns) of the dataset.	10001	100001	1618	501
Number of distinct values of the target attribute (if it is nominal).	2	2	2	2
Number of missing values in the dataset.	0	0	0	0
Number of instances with at least one value missing.	0	0	0	0
Number of numeric attributes	10000	100000	1617	500
Number of nominal attributes	1	1	1	1
Percentage of binary attributes	0	0	0.06	0.2
Percentage of instances having missing values	0	0	0	0
Average class difference between consecutive instances	0.44	0.82	0.93	0.51
Percentage of missing values	0	0	0	0
Number of attributes divided by the number of instances	50.01	0.9	0.38	0.19
Percentage of numeric attributes	1	1	99.94	99.8
Percentage of instances belonging to the most-frequent class	0.56	0.9	96.48	50
Percentage of nominal attributes	0	0	0.06	0.2
Number of instances belonging to the most-frequent class	112	1038	4080	1300
Percentage of instances belonging to the least-frequent class	0.44	0.1	0.04	0.5
Number of instances belonging to the least-frequent class	88	112	149	1300
Number of binary attributes	1	1	1	1

2) DOROTHEA is a drug-discovery dataset [103][75]. Chemical compounds, represented by structural molecular features, must be classified as active (binding to thrombin) or inactive. This is one of five datasets of the NIPS 2003 feature selection challenge.

3) HIVA or Hiva agnostic [75] is a part of the Agnostic Learning vs. Prior Knowledge Challenge. HIVA is the HIV infection database. HIVA originally had three classes

(active, moderately active, and inactive), but in this research, it used the two-class classification problem (active vs. inactive).

4) MADELON is an artificial dataset [103][75] which was part of the NIPS 2003 feature selection challenge. This is a two-class classification problem with continuous input variables. This dataset is one of five datasets used in the NIPS 2003 feature selection challenge.

5.2 Classification Algorithms

Four learning algorithms are used to evaluate the classification performance: decision tree, random forest, support vector machine (SVM), and K-nearest-neighbor (KNN). These learning algorithms are used because of their popularity in the recently published literature as well as their ranking as the most-accurate [104] data-mining algorithms.

- Decision Tree [104] is a type of supervised-learning algorithm that is mostly used in used in statistics, data mining and machine learning. In classification problems decision tree would be the first choice for prediction modelling approach to be select. In this technique, the data is split into two or more homogeneous sets based on the most significant splitter differentiator in input variables. In this work, the C4.5 algorithm is used.

- Random forest [105] is one of the common algorithms that is considered for classification problem. It is a classification method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification). Random forest creates multiple decision trees and merges them together to get a better stable and accurate prediction result. At this work, since this work have five datasets, it tried to use different learning algorithms for performing classification.

- Support Vector Machines (SVM) is one of the most popular algorithms for large-margin classification [104]. The idea of the SVM algorithm is to map the given training set into a possibly high-dimensional feature space and attempting to locate in that space a hyperplane that maximizes the distance separating the positive from the negative examples. Having found such a hyperplane, the SVM can then predict the classification of an unlabeled example.

- K-Nearest-Neighbor (KNN) [104] is one of the simplest and most trivial classifiers; KNN is a non-parametric method, and in its classification it employs k, which is the number of its nearest neighbours, to classify data to its groups; it provides good generalization accuracy on many domains, learns very quickly, and is easy to understand. On the other hand, the KNN algorithm has large storage requirements because it has to store all of the data; it is slow with large datasets because all of the training instances have to be visited. The accuracy of the NN algorithm degrades with an increase of noise and irrelevant attributes in the training data.

5.3 Competing Approaches

SFGS approach is a single streaming feature selection, thus three competing approaches are choose from Chapter 2 to compare the algorithms' performance as follows:

- Alpha-investing [44] which is one of the earliest well-known proposed algorithms in this area.

- Fast-Online Streaming Feature Selection (Fast-OSFS) [45] which is one of the most recent state-of-arts streaming feature selection algorithms.

- Predominant group-based variable neighborhood search (PGVNS) Algorithm [26].

5.4 Hardware and Software Environment

The experiment is conducted on a computer with Windows 10, an Intel Core i7 processor, 1TB SSD, and 32GB RAM. The proposed algorithm is developed on NetBeans IDE 8.2.

5.5 Parameters Setup

- The parameter q: is used to determine if the new candidate feature will be added to one of the existing groups.

- Parameters for learning algorithms: as the objective of this work is nominating the best informative features, optimizing learners' parameters to provide the best classification accuracy is not the aim here. Thus, this work used the default setting for each learner through all the experiment as follows:

    - Decision Tree: C4.5 version. The number of folds is 10. Confidence factor value is 25%. The minimum number of instances in the two most popular branches (default 2).

    - Random forest: Number of folds is 10. The number of iterations is 100. The batch size is 100. The seeds is one. The number of iterations is 100.

    - Support Vector Machines (SVM): The number of folds is 10. The batch size is 100.

  - K-Nearest-Neighbor (KNN): The number of folds is 10. The batch size is 100. The k is 1.

- Gain Ratio thresholds: at the relevance test, the SFGS used different values (0.01, 0.05 and 0.1) to examine new coming features. Feature weighting is used to improve classification accuracy by discarding non-informative features with weights below a certain threshold value. Thus, it can increase the resource efficiency of the classifier and handle the big data criteria.

- Initial dataset: in this experiment, it is assumed that initial datasets are half of the original features in each dataset.

5.6 Results and Evaluation

In the experimental work, a 10-fold cross-validation is used to evaluate the quality of the final features' subset selection. It preferred setting to compare the prediction accuracy of the proposed algorithm with the three competing algorithms PGVNS, Fast-OSFS and Alpha-investing. Thus, it allows for more efficient use of the data and provides a more-accurate estimation of out-of-sample accuracy. As a summary, Table 17 shows the resulted accuracy for the five datasets. Additionally, Figure 25 to Figure 28 illustrate the accuracy results, in which the accuracy of the proposed SFGS is always better than PGVNS, Alpha and clearly can well compete with fast-OSFS.

5.6.1 Proposed SFGS vs Fast-OSFS

Comparing the results between the proposed SFGS with the fast-OSFS, the SFGS has better performance in three datasets Arcene, Dorothea and Madelon. Besides that, SFGS has almost the same accuracy results in Hiva agnostic dataset according to Table 17. For example, in Dorothea dataset, fast-OSFS achieves 92.80% as accuracy for decision tree learning algorithm. By contrast, the proposed SFGS achieves 96.42% with Dorothea dataset. Dorothea dataset consists of one hundred thousand features, which consider the largest dataset on the UCI in term of number of features. Fast-OSFS has two versions of implementation. The first one is FAST_OSFS_D for discrete data using Chi-square. The second version is FAST_OSFS_Z for numerical data using Fisher's Z test. Fast_OSFS uses the ranking at the relevance test with threshold either 0.01 or 0.05, which reflects that most of the features are considered as not relevant.

5.6.2 Proposed SFGS vs Alpha-Investing

Comparing the results between the proposed SFGS with alpha-investing, the SFGS has better performance in the all datasets. For example, Arcene dataset has ten-thousand features, SFGS achieves 88.57% for KNN accuracy, whereas alpha-investing achieves only 72.86%. Another scenario is the decision tree. SFGS achieves 76.43% whereas alpha-investing achieves 70.71%. Please refer to Table 17.

5.6.3 Proposed SFGS vs PGVNS

Also comparing the accuracy of grouping criteria between SFGS and PGVNS as Table 17 reports. Comparing the accuracy results between SFGS with PGVNS, the SFGS achieves the highest accuracy in the four datasets, where the difference is visible. For example, comparing decision tree average results in the Arcene dataset, PGVNS achieves 70.71%. By contrast, SFGS achieves 76.43%. Similar performance is observed for the other classifiers. Consequently, it could interpret the higher accuracy prediction results of SFGS due to the better feature selection. Streaming the features and updating the model each time allows proposed SFGS to better utilize the informative features by using the grouping approach.

Table 17: Shows the accuracy results of the four datasets using four learning algorithms with CV fold-10. The last column summarizes the average accuracy of each approach on the declared dataset, where the proposed SFGS shows better performance. Likewise, the last raw presents the average accuracy results of each approach using the declared learning accuracy.

	Approaches	**Decision tree learning algorithm**	**Random forest learning algorithm**	**Support vector machines (SVM) learning algorithm**	**K-nearest-neighbor (KNN) learning algorithm**	**The overall accuracy average**
Arcene	PGVNS	71%	68%	78%	74%	73%
	Proposed SFGS	**76%**	**83%**	**86%**	**89%**	**83%**
	FAST OSFS	69%	76%	77%	70%	73%
	Alpha Investing	71%	71%	79%	73%	74%
	All features	71%	82%	85%	87%	81%
Dorothea	PGVNS	89%	89%	90%	89%	89%
	Proposed SFGS	**96%**	**97%**	**96%**	**97%**	**97%**
	FAST OSFS	93%	95%	95%	95%	95%
	Alpha Investing	90%	94%	95%	90%	92%
	All features	92%	90%	92%	89%	91%
Hiva-agnostic	PGVNS	96%	96%	97%	96%	97%
	Proposed SFGS	96%	96%	96%	96%	96%
	FAST OSFS	97%	97%	97%	97%	97%
	Alpha Investing	95%	97%	97%	95%	96%
	All features	96%	97%	94%	95%	96%
Madelon	PGVNS	50%	50%	51%	50%	50%
	Proposed SFGS	**66%**	**68%**	**60%**	**62%**	**64%**
	FAST OSFS	60%	57%	60%	55%	58%
	Alpha Investing	60%	56%	60%	53%	57%
	All features	67%	61%	54%	54%	59%
The overall accuracy average	PGVNS	77%	76%	79%	77%	77%
	Proposed SFGS	**84%**	**86%**	**85%**	**86%**	**85%**
	FAST OSFS	80%	81%	82%	79%	81%
	Alpha Investing	79%	80%	83%	78%	80%
	All features	82%	83%	81%	81%	82%

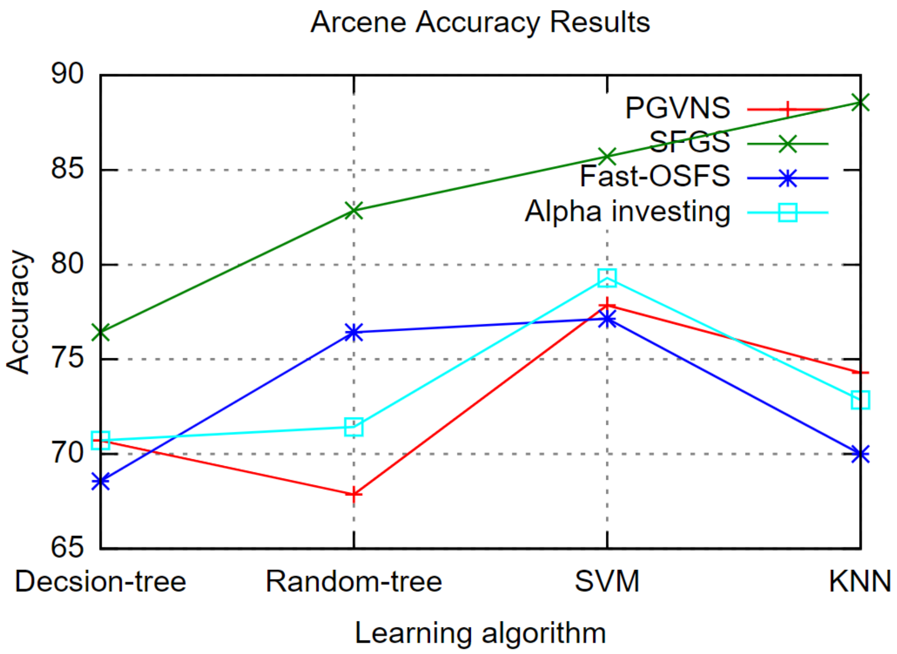

Figure 25: Arcene accuracy's results show the performance of the SFGS comparing to the other competing approaches. SFGS presents high performance in the accuracy of the four learning algorithms. The highest accuracy result is the KNN CV fold-10 where it achieves 88.57%.

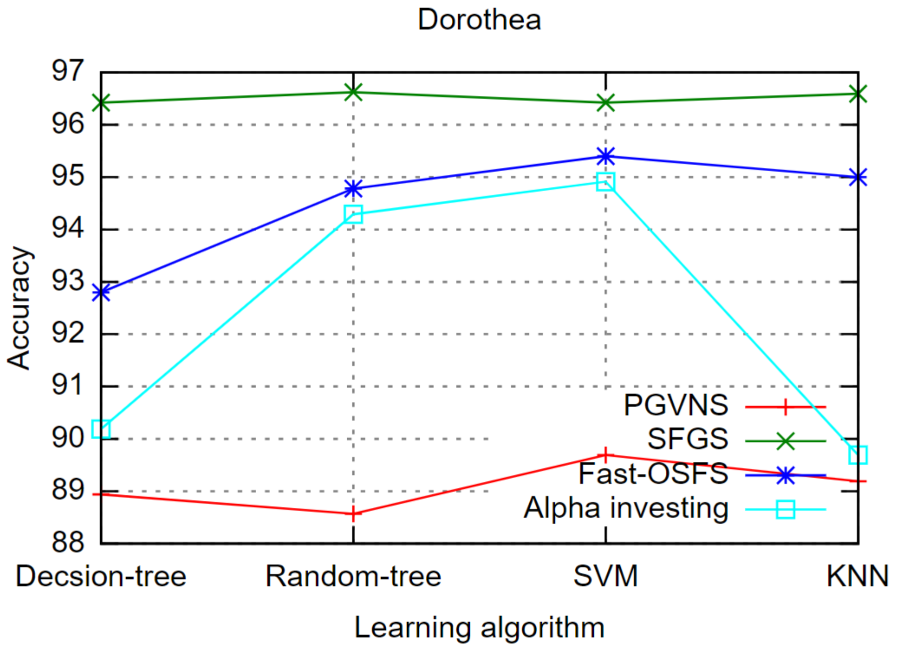

Figure 26: Dorothea accuracy's results also shows the highest performance of the SFGS comparing to the other competing approaches. SFGS achieves around 96% for the four learning algorithms.

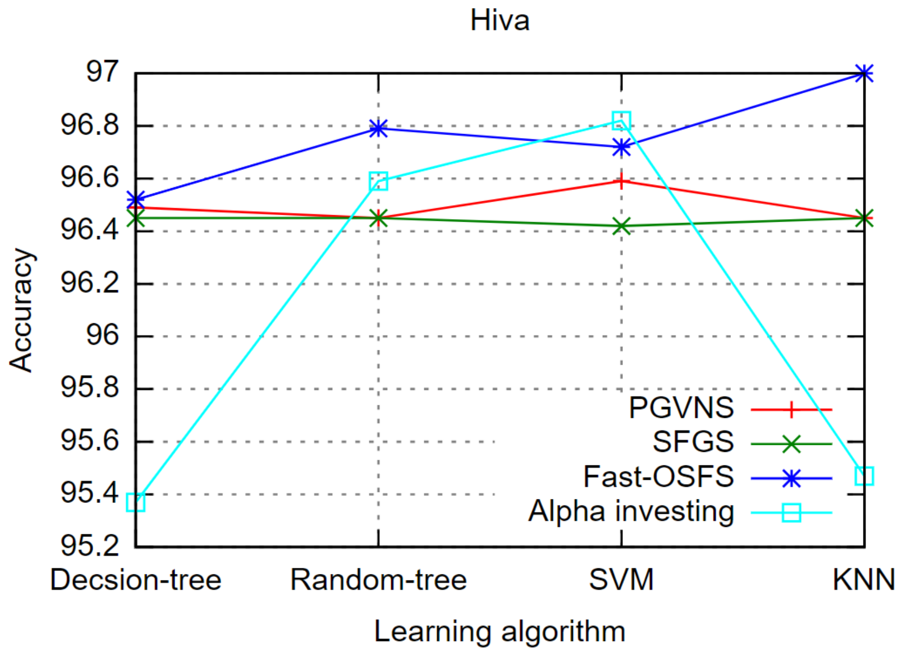

Figure 27: Hiva accuracy's results shows the much close performance for the four competing approaches. The SFGS presents more stable performance among the other competing approaches.

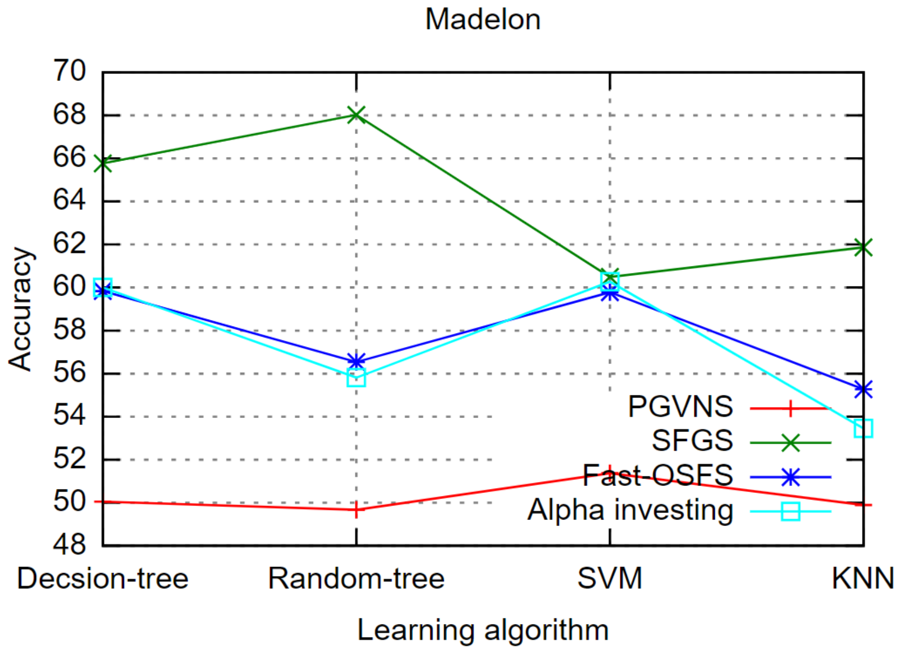

Figure 28: Madelon accuracy's results shows the highest performance of the SFGS comparing to the other competing approaches. SFGS presents high performance in the accuracy of the four learning algorithms. The highest accuracy result is the random forest CV fold-10 where it achieves 68.02%.

5.6.4 Running Time Performance (CPU Time) Analysis

In addition to the classification accuracy, the execution time measure is considered to evaluate the performance of the proposed algorithm against the three competing approaches. Figure 29 illustrates the resulting summary for the running-time comparison between proposed SFGS and the competing approaches, PGVNS Alpha-investing and Fast-OSFS.

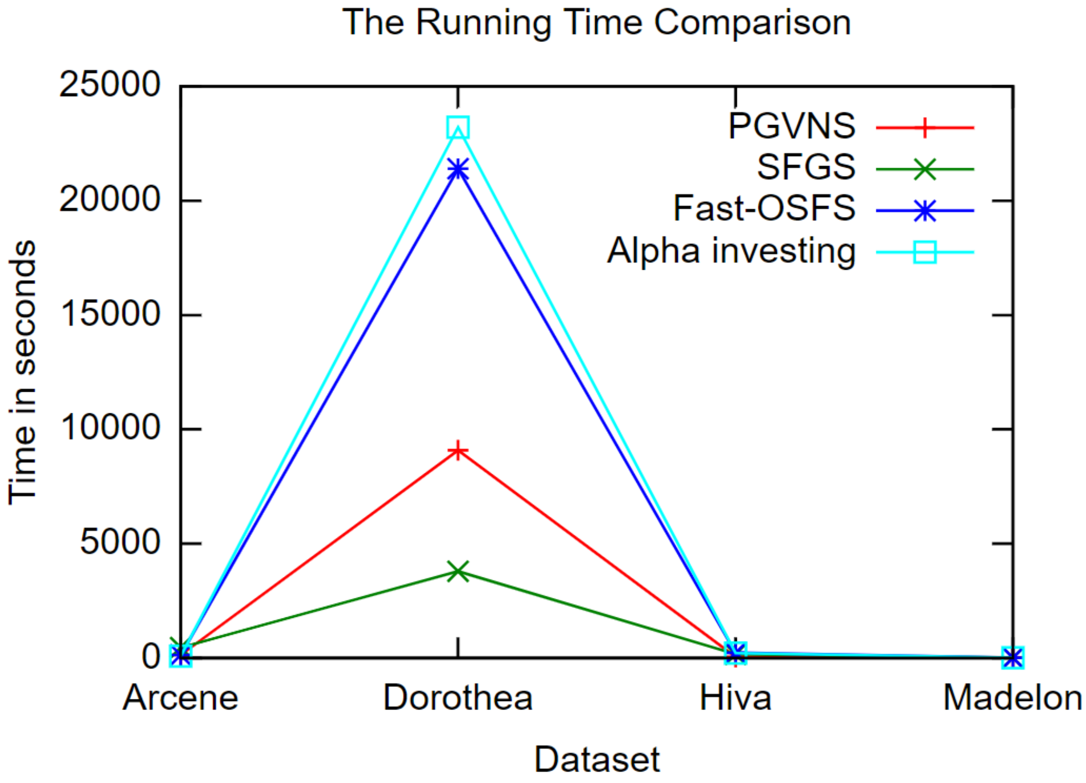

Figure 29: Running time comparison of the four competing approaches. The time presents the streaming feature selection timing and the building the model to report the accuracy. In the SFGS case, the model is updated each time there is a new feature.

The running time comparison includes streaming feature selection and building the four learning models to report the accuracy results. Note that PGVNS, Alpha-investing and Fast-OSFS are only feature selection algorithms. Thus, it carried out building the four learning models in order to report prediction accuracy. Referring to Figure 30, there are some specific cases where the time is not reported. For example, running time of both PGVNS and Fast-OSFS on Internet advertisements dataset is not included, because the accuracy result is always zero. This is because the algorithms failed to process the data. The ranking result at the relevancy test is always zero. Thus, all the features were excluded. Thus, it didn’t report these cases. Figure 24 in Section 4.2 illustrates the running time comparison, where the lower graph indicates better performance. The SFGS has better performance because of the grouping strategy. So, instead of comparing the coming features with the previously accepted features. SFGS compares the upcoming feature with the informative features from each group. Thus, it reduces the processing time in a better way compared to the other approaches.

5.6.5 Parameter Sensitivity

Sensitivity analysis is particularly valuable in obtaining certainty in the results of the primary analysis. This section studies the effect of the SFGS approach's parameters on the final accuracy results by changing one input and keeping the others constant. These parameters are defined in Section 5.5, "Parameters Setup": (1) sensitivity to the gainRatio threshold, (2) sensitivity to parameter q and (3) sensitivity to initial groups. The outcome of sensitivity analysis can have important implications for the SFGS approach by investigating more broadly the relationship between these parameters and the final learning model.

Sensitivity to GainRatio Threshold

In the gainRatio sensitivity experiment, it examined the effect of various values of the gainRatio threshold in the relevance test stage to examine new, coming features. This investigation seeks a balance between the features gained and grouping quality. By allowing for more features, the grouping strategy would have better performance in selecting the most informative features. However, it also want to reduce these new, coming features. Therefore, possible values are applied to the biggest dataset, Dorothea. Table 18 shows the number of the accepted feature from the feature stream.

Table 18: Relevancy test using different gainratio thresholds

GainRatio Threshold	0.01	0.03	0.05	0.1	0.3	0.33	0.34	0.35
Number of relevance features	5867	5207	5130	5050	913	224	101	45

Threshold values between 0.01 and 0.1 would indicate more relevant features that could generate more single groups. In contrast, a threshold value higher than 0.30 would generate fewer single groups and force new relevant features to join one of the existing groups. The best group quality would result from a threshold value between

0.1 and 0.3. More discussion about the quality of the resulting groups will follow in the next section.

Sensitivity to Parameter q

In the parameter q experiment, it attempts to determine the best balance between the attributes' quality distribution and the resulting accuracy. In this experiment, it examines various values for the q. Table 19 shows the totals of the two types of groups: (a) single attribute group and (b) group(s) with two or more attributes. Choosing a q value between 0.001 and 0.01 would generate more single attribute groups than groups with 2 or more attributes. In this approach, the final subset of selected features consists of the representative feature from each group. Therefore, the total number of representative features will increase by a single feature group. Therefore, the high dimensionality of the streaming feature would probably not be reduced. In contrast, values greater than 0.05 would generate fewer single feature groups, but the total number of groups will remain constant. The best value of q would be 0.015. In addition to grouping quality, Table 20 shows the trade-off between accuracy results of q values.

Table 19: The relation between the q and the quality of the resulted groups on Dorothea dataset

q	**0.001**	**0.005**	**0.01**	**0.015**	**0.05**	**2**
Total groups	160	155	151	78	72	72
Single attribute group(s)	98	93	89	16	10	10
Group(s) with more than 2 attributes	62	62	62	62	62	62
Group 0	2400	2400	2400	2400	2400	2400
Group 1	314	314	314	314	314	314
Group 2	99	100	100	100	100	100
Group 3	799	799	799	824	825	825
Group 4	558	558	558	559	566	566
Group 5	476	476	476	476	476	476
Group 6	20	20	20	20	20	20
Group 7	147	147	147	147	147	147
Group 8	110	110	110	111	111	111
Group 9	263	263	263	263	263	263
Group 10	4	4	4	4	4	4
Group 11	32	32	32	32	32	32
Group 12	82	82	82	82	82	82
Group 13	22	22	22	22	22	22
Group 14	39	39	39	39	39	39
Group 15	18	18	19	19	19	19
Group 16	12	12	12	13	14	14
Group 17	11	11	11	12	12	12
Group 18	10	11	11	11	11	11
Group 19	25	25	26	26	26	26
Group 20	16	16	16	17	17	17
Group 21	9	9	9	9	9	9
Group 22	4	5	5	5	5	5
Group 23	7	7	8	8	8	8
Group 24	6	6	6	20	7	7
Group 25	4	4	4	4	4	4
Group 26	15	15	15	16	26	26
Group 27	3	3	3	11	4	4
Group 28	3	3	4	4	4	4
Group 29	8	8	2	9	9	9
Group 30	4	4	5	4	4	4
Group 31	2	2	2	2	2	2
Group 32	5	5	5	7	6	6
Group 33	4	4	4	4	4	4
Group 34	1	1	1	1	1	1
Group 35	4	4	4	4	4	4
Group 36	6	6	6	6	6	6

Table 19: The relation between the q and the quality of the resulted groups on Dorothea dataset (continued)

q	**0.001**	**0.005**	**0.01**	**0.015**	**0.05**	**2**
Group 37	12	12	12	12	12	12
Group 38	5	5	5	5	5	5
Group 39	3	3	3	3	3	3
Group 40	5	5	5	5	5	5
Group 41	2	2	2	2	2	2
Group 42	1	1	1	1	1	1
Group 43	2	2	2	3	3	3
Group 44	6	6	6	6	6	6
Group 45	122	122	122	122	122	122
Group 46	46	46	46	51	47	47
Group 47	4	4	4	7	6	6
Group 48	7	7	7	8	16	16
Group 49	5	5	5	5	5	5
Group 50	7	7	7	7	7	7
Group 51	2	2	2	4	7	7
Group 52	2	2	2	2	2	2
Group 53	0	0	0	1	1	1
Group 54	3	2	3	4	4	4
Group 55	5	5	5	5	5	5
Group 56	1	1	1	1	1	1
Group 57	3	3	2	4	5	5
Group 58	2	2	2	2	2	2
Group 59	2	2	2	3	3	3
Group 60	3	3	3	3	4	4
Group 61	2	0	0	0	0	0
Group 62	1	1	2	2	2	2
Group 63	0	1	1	1	1	1
Group 64	0	1	1	1	1	1
Group 65	0	1	1	1	1	1
Group 66	0	1	1	1	1	1
Group 67	0	1	1	1	1	1
Group 68	0	1	1	1	1	1
Group 69	0	0	0	1	0	0
Group 70	0	0	0	1	0	0
Group 71	0	0	0	1	1	0
Group 72	0	0	0	1		
Group 73	0	0	0	1		
Group 74	0	0	0	0		
Group 75	0	0	0			
Group 76	0	0	0			
Group 77	0	0	0			

Table 19: The relation between the q and the quality of the resulted groups on Dorothea dataset (continued)

q	**0.001**	**0.005**	**0.01**	**0.015**	**0.05**	**2**
Group 78	0	0	0			
Group 79	0	0				
Group 80	0	0				
Group 81	0	0				
Group 82	0	0				
Group 83	0	0				
Group 84	0	0				
Group 85	0	0				
Group 86	0	0				
Group 87	0	0				
Group 88	0	0				
Group 89	0	0				
Group 90	0	0				
Group 91	0	0				
Group 92	0	0				
Group 93	0	0				
Group 94	0	0				
Group 95	0	0				
Group 96	0	0				
Group 97	0	0				
Group 98	0	0				
Group 99	0	0				
Group 100	0	0				
Group 101	0	0				
Group 102	0	0				
Group 103	0	0				
Group 104	0	0				
Group 105	0	0				
Group 106	0	0				
Group 107	0	0				
Group 108	0	0				
Group 109	0	0				
Group 110	0	0				
Group 111	0	0				
Group 112	0	0				
Group 113	0	0				
Group 114	0	0				
Group 115	0	0				
Group 116	0	0				
Group 117	0	0				
Group 118	0	0				
Group 119	0	0				
Group 120	0	0				

Table 19: The relation between the q and the quality of the resulted groups on Dorothea dataset (continued)

q	**0.001**	**0.005**	**0.01**	**0.015**	**0.05**	**2**
Group 121	0	0				
Group 122	0	0				
Group 123	0	0				
Group 124	0	0				
Group 125	0	0				
Group 126	0	0				
Group 127	0	0				
Group 128	0	0				
Group 129	0	0				
Group 130	0	0				
Group 131	0	0				
Group 132	0	0				
Group 133	0	0				
Group 134	0	0				
Group 135	0	0				
Group 136	0	0				
Group 137	0	0				
Group 138	0	0				
Group 139	0	0				
Group 140	0	0				
Group 141	0	0				
Group 142	0	0				
Group 143	0	0				
Group 144	0	0				
Group 145	0	0				
Group 146	0	0				
Group 147	0	0				
Group 148	0	0				
Group 149	0	0				
Group 150	0	0				
Group 151	0	0				
Group 152	0	0				
Group 153	0	0				
Group 154	0	0				
Group 155	0	0				
Group 156	0					
Group 157	0					
Group 158	0					
Group 159	0					
Group 160	0					
Groups average features	**84**	**84**	**84**	**85**	**85**	**85**

Table 20: The relation between q, and the resulted accuracy on Dorothea dataset

q	Learning Algorithm	Accuracy	TP Rate	FP Rate	Precision	Recall
0.001	**Decision Tree**	92.92%	92.92%	46.93%	92.39%	92.92%
	Random Forest	93.29%	93.29%	42.09%	92.78%	93.29%
	SMO	96.02%	96.02%	25.44%	95.90%	96.02%
	KNN	93.17%	93.17%	48.81%	92.93%	93.17%
0.005	**Decision Tree**	92.92%	92.92%	46.93%	92.39%	92.92%
	Random Forest	94.41%	94.41%	35.23%	94.10%	94.41%
	SMO	96.02%	96.02%	25.44%	95.90%	96.02%
	KNN	93.29%	93.29%	47.84%	93.06%	93.29%
0.01	**Decision Tree**	92.92%	92.92%	46.93%	92.39%	92.92%
	Random Forest	94.04%	94.04%	37.20%	93.66%	94.04%
	SMO	96.02%	96.02%	25.44%	95.90%	96.02%
	KNN	93.29%	93.29%	47.84%	93.06%	93.29%
0.015	**Decision Tree**	92.80%	92.80%	47.90%	92.24%	92.80%
	Random Forest	94.04%	94.04%	38.15%	93.68%	94.04%
	SMO	95.65%	95.65%	29.32%	95.54%	95.65%
	KNN	92.92%	92.92%	52.68%	92.89%	92.92%
0.05	**Decision Tree**	92.92%	92.92%	46.93%	92.39%	92.92%
	Random Forest	94.16%	94.16%	37.18%	93.82%	94.16%
	SMO	95.28%	95.28%	28.41%	95.06%	95.28%
	KNN	92.92%	92.92%	52.68%	92.89%	92.92%
2	**Decision Tree**	92.92%	92.92%	46.93%	92.39%	92.92%
	Random Forest	94.16%	94.16%	37.18%	93.82%	94.16%
	SMO	95.28%	94.16%	37.18%	93.82%	94.16%
	KNN	92.92%	92.92%	52.68%	92.89%	92.92%

Sensitivity to Initial Groups

The initial grouping experiment is aimed to study the relationship between the learning accuracy result and the initial groups. Note that the competing approaches are not sensitive to randomization, nor is this approach. Each time it randomize the dataset, it get different initial groups. Therefore, the results may slightly differ, but they are still better than those from the competing approaches. Table 21 shows the comparison of the randomized Dorothea dataset and other approaches.

Table 21: Random initial grouping comparison between SFGS and other competing approaches

Approaches	**Decision tree learning algorithm**	**Random forest learning algorithm**	**Support vector machines (SVM) learning algorithm**	**K-nearest-neighbor (KNN) learning algorithm**
PGVNS	89%	89%	90%	89%
The proposed SFGS	**96%**	**97%**	**96%**	**97%**
FAST OSFS	93%	95%	95%	95%
Alpha Investing	90%	94%	95%	90%
All features	92.30%	89.69%	92.17%	89.07%
Random initial groups average	**94%**	**95%**	**95%**	**95%**

In the next chapter, a conclusion and future work are presented.

Chapter 6: Conclusion and Future Research

6.1 Conclusion

This dissertation aimed to address the challenges of streaming feature selection for classification problems in big data. This work focused on feature selection methods for problems in which the total number of predictive features is challenging to determine from a large set of potential features. In the worst cases, the total number of features is unknown.

To achieve the research aim, the dissertation utilized the feature grouping principle as a powerful approach to resolve the issue of big data volume. Feature grouping selects relevant features by measuring the hidden information between the selected features and nominating the most informative ones in the group. This process allows us to develop a streaming feature grouping and selection (SFGS) algorithm to resolve this issue. SFGS integrates online feature selection and feature grouping into one framework, which it is called streaming feature grouping.

This dissertation makes three main contributions: First, it delivered the novel SFGS technique. Second, it addressed the challenge of reducing the extremely high dimensionality when classifying features in big data. Third, the SFGS can be integrated into real-world applications that manipulate real-time data. Thus, it could support these applications' ability to more efficiently yield real-time analytics. Finally, the SFGS has been evaluated with real data and compared to state-of-the-art algorithms.

6.2 Recommendations and Future Research

This dissertation makes several recommendations to improve and extend the SFGS approach. Future researchers could focus on these areas to further improve and expand the work of this dissertation.

The SFGS approach is developed on grouping criteria. Similar features are arranged in one group, and dissimilar features are arranged in another group. The two resulting groups are single attribute groups and groups with more than two attributes. More investigation needs to be conducted on the relationship between group size and resulting accuracy. For example, a trade-off between the effect of a threshold on each group's size and the resulting groups would be needed. Putting a threshold on each group's quality in terms of intra-cluster cohesion and inter-cluster dispersion would lead to more accurate results. A hierarchical clustering approach would also be an area for investigation, merging single groups into a large one or splitting a large group into smaller ones. The critical point is how to balance accuracy and group stability.

Additionally, applying further performance measures to evaluate the experimental work because the classification may give satisfying results when it is evaluated using only one metric, such as an accuracy score. However, using another metric, such as the area under the curve, may give an unsatisfying result. Most of the researchers use classification accuracy to evaluate their results; however, it is not enough to truly judge the result.

On the other hand, working on larger datasets will lead to a better understanding of the limitations of SFGS. Streaming feature selection focus concerns only the number of features. However, testing datasets with larger dimensionality in terms of features and instances would open the door for future improvement.

Moreover, the current streaming feature selection approaches are used as supervised learning problems to select the most informative features. However, more investigation of how streaming feature selection could be applied to unsupervised learning is needed.

The SFGS approach introduced in this dissertation acts as a guide to future research. In recent years, most research in the streaming feature selection domain has only focused on the feature stream. Likewise, data stream selection is also focused on data selection. In contrast, handling data streaming and feature streaming simultaneously would open a direction for future research. Big data is generated from many sources, and a huge demand has arisen for real-time analysis.

References

[1] G. Press, “6 Predictions For The $203 Billion Big Data Analytics Market,” *Forbes*, 20-Jan-2017. [Online]. Available: https://www.forbes.com/sites/gilpress/2017/01/20/6-predictions-for-the-203-billion-big-data-analytics-market/#6b26d76c2083.

[2] S. C. Hoi, J. Wang, P. Zhao, and R. Jin, “Online feature selection for mining big data,” in *Proceedings of the 1st international workshop on big data, streams and heterogeneous source mining: Algorithms, systems, programming models and applications*, 2012, pp. 93–100.

[3] L. Landberg, G. Giebel, H. A. Nielsen, T. Nielsen, and H. Madsen, “Short-term prediction—an overview,” *Wind Energy Int. J. Prog. Appl. Wind Power Convers. Technol.*, vol. 6, no. 3, pp. 273–280, 2003.

[4] K. Yu, W. Ding, D. A. Simovici, H. Wang, J. Pei, and X. Wu, “Classification with streaming features: An emerging-pattern mining approach,” *ACM Trans. Knowl. Discov. Data TKDD*, vol. 9, no. 4, p. 30, 2015.

[5] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant Features and the Subset Selection Problem,” in *Machine Learning Proceedings 1994*, W. W. Cohen and H. Hirsh, Eds. San Francisco (CA): Morgan Kaufmann, 1994, pp. 121–129.

[6] L. Yu and H. Liu, “Efficient Feature Selection via Analysis of Relevance and Redundancy,” *J Mach Learn Res*, vol. 5, no. 10, pp. 1205–1224, Dec. 2004.

[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 112. Springer, 2013.

[8] N. T. Longford, “A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects,” *Biometrika*, vol. 74, no. 4, pp. 817–827, 1987.

[9] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[10] D. Koller and M. Sahami, “Toward Optimal Feature Selection,” in *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, San Francisco, CA, USA, 1996, pp. 284–292.

[11] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.

[12] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 856–863.

[13] K. Kira and L. A. Rendell, “A practical approach to feature selection,” in *Proceedings of the ninth international workshop on Machine learning*, 1992, pp. 249–256.

[14] M. Robnik-Šikonja and I. Kononenko, “Theoretical and empirical analysis of ReliefF and RReliefF,” *Mach. Learn.*, vol. 53, no. 1–2, pp. 23–69, 2003.

[15] Q. Gu, Z. Li, and J. Han, “Generalized Fisher Score for Feature Selection,” *CoRR*, vol. abs/1202.3725, 2012.

[16] D. D. Lewis, “Feature selection and feature extraction for text categorization,” in *Proceedings of the workshop on Speech and Natural Language*, 1992, pp. 212–217.

[17] R. Battiti, “Using mutual information for selecting features in supervised neural net learning,” *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, 1994.

[18] D. Lin and X. Tang, “Conditional infomax learning: an integrated framework for feature extraction and fusion,” *Comput. Vision–ECCV 2006*, pp. 68–82, 2006.

[19] H. H. Yang and J. Moody, “Data visualization and feature selection: New algorithms for nongaussian data,” in *Proceedings of Advances in Neural Information Processing Systems*, 2000, pp. 687–693.

[20] F. Fleuret, “Fast binary feature selection with conditional mutual information,” *J. Mach. Learn. Res.*, vol. 5, no. 10, pp. 1531–1555, 2004.

[21] M. Vidal-Naquet and S. Ullman, “Object Recognition with Informative Features and Linear Classification.,” in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, vol. 3, pp. 281–288.

[22] A. Jakulin, “Machine learning based on attribute interactions,” Univerza v Ljubljani, 2005.

[23] P. E. Meyer and G. Bontempi, “On the use of variable complementarity for feature selection in cancer classification,” in *Proceedings Workshops on Applications of Evolutionary Computation*, 2006, pp. 91–102.

[24] K. Kira and L. A. Rendell, “The feature selection problem: Traditional methods and a new algorithm,” in *Proceedings of the Tenth National Conference on Artificial Intelligence*, 1992, vol. 2, pp. 129–134.

[25] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, “Overcoming the myopia of inductive learning algorithms with RELIEFF,” *Appl. Intell.*, vol. 7, no. 1, pp. 39–55, 1997.

[26] M. García-Torres, F. Gómez-Vela, B. Melián-Batista, and J. M. Moreno-Vega, “High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach,” *Inf. Sci.*, vol. 326, pp. 102–118, 2016.

[27] Q. Song, J. Ni, and G. Wang, “A fast clustering-based feature subset selection algorithm for high-dimensional data,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 1–14, 2013.

[28] H. D. Gangurde, “Feature Selection using Clustering approach for Big Data,” *Int. J. Comput. Appl.*, vol. 975, pp. 1–3, 2014.

[29] H. D. Gangurde and K. V. Metre, “Mining of High Dimensional Data using Feature Selection,” *Int. J. Comput. Sci. Mob. Comput.*, vol. 4, no. 6, pp. 901–906, Jun. 2015.

[30] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997.

[31] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Amsterdam: Elsevier, 2011.

[32] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. R. Stat. Soc. Ser. B Methodol.*, pp. 267–288, 1996.

[33] M. Yamada, W. Jitkrittum, L. Sigal, E. Xing, and M. Sugiyama, “High-dimensional feature selection by feature-wise non-linear lasso. arXiv preprint,” *ArXiv Prepr. ArXiv12020515*, 2012.

[34] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, 2005.

[35] C. C. Aggarwal, *Data classification: algorithms and applications*. CRC Press, 2014.

[36] J. Ye and J. Liu, “Sparse methods for biomedical data,” *ACM Sigkdd Explor. Newsl.*, vol. 14, no. 1, pp. 4–15, 2012.

[37] S. Kim and E. P. Xing, “Statistical Estimation of Correlated Genome Associations to a Quantitative Trait Network,” *PLoS Genet.*, vol. 5, no. 8, pp. 1–18, 2009.

[38] S. Yang, L. Yuan, Y.-C. Lai, X. Shen, P. Wonka, and J. Ye, “Feature grouping and selection over an undirected graph,” in *Graph Embedding for Pattern Analysis*, Springer, 2013, pp. 27–43.

[39] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 68, no. 1, pp. 49–67, 2006.

[40] J. H. Friedman, T. J. Hastie, and R. Tibshirani, “A note on the group lasso and a sparse group lasso,” 2010.

[41] J. Liu and J. Ye, “Moreau-Yosida regularization for grouped tree structure learning,” in *Proceedings of Advances in Neural Information Processing Systems*, 2010, pp. 1459–1467.

[42] J. Tang, S. Alelyani, and H. Liu, “Feature selection for classification: A review,” *Data Classif. Algorithms Appl.*, p. 37, 2014.

[43] S. Perkins and J. Theiler, “Online feature selection using grafting,” in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 592–599.

[44] J. Zhou, D. Foster, R. Stine, and L. Ungar, “Streaming feature selection using alpha-investing,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 384–393.

[45] X. Wu, K. Yu, H. Wang, and W. Ding, “Online streaming feature selection,” in *Proceedings of 27th international conference on machine learning (ICML-10)*, 2010, pp. 1159–1166.

[46] S. Eskandari and M. Javidi, “Online streaming feature selection using rough sets,” *Int. J. Approx. Reason.*, vol. 69, pp. 35–57, 2016.

[47] “What is Big Data Analytics?” [Online]. Available: http://www-01.ibm.com/software/data/infosphere/hadoop/what-is-big-data-analytics.html.

[48] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, “Online feature selection with streaming features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1178–1192, 2013.

[49] K. Yu, X. Wu, W. Ding, and J. Pei, “Towards Scalable and Accurate Online Feature Selection for Big Data,” in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2014, pp. 660–669.

[50] F. Wang, J. Liang, and Y. Qian, “Attribute reduction: a dimension incremental strategy,” *Knowl.-Based Syst.*, vol. 39, pp. 95–108, 2013.

[51] M. M. Javidi and S. Eskandari, “Streamwise feature selection: A rough set method,” *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 4, pp. 667–676, 2018.

[52] A. Tommasel and D. Godoy, “A Social-aware online short-text feature selection technique for social media,” *Inf. Fusion*, vol. 40, pp. 1–17, 2018.

[53] P. Zhou, X. Hu, P. Li, and X. Wu, “OFS-Density: A novel online streaming feature selection method,” *Pattern Recognit.*, vol. 86, pp. 48–61, 2019.

[54] H. Li, X. Wu, Z. Li, and W. Ding, “Group feature selection with streaming features,” in *Proceedings of 13th International Conference on Data Mining*, 2013, pp. 1109–1114.

[55] K. Yu, X. Wu, W. Ding, and J. Pei, “Scalable and accurate online feature selection for big data,” *ACM Trans. Knowl. Discov. Data TKDD*, vol. 11, no. 2, p. 16, 2016.

[56] Jing Wang *et al.*, “Online feature selection with group structure analysis,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3029–3041, 2015.

[57] J. Wang, Z.-Q. Zhao, X. Hu, Y.-M. Cheung, M. Wang, and X. Wu, "Online Group Feature Selection.," in *IJCAI*, 2013, pp. 1757–1763.

[58] K. Yu, W. Ding, and X. Wu, "LOFS: A library of online streaming feature selection," *Knowl.-Based Syst.*, vol. 113, pp. 1–3, 2016.

[59] Y. Zhuang, K. Yu, D. Wang, and W. Ding, "An evaluation of big data analytics in feature selection for long-lead extreme floods forecasting," in *Proceedings 2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)*, 2016, pp. 1–6.

[60] B. Auffarth, M. López, and J. Cerquides, "Comparison of Redundancy and Relevance Measures for Feature Selection in Tissue Classification of CT Images.," in *Proceedings of the ICDM*, 2010, pp. 248–262.

[61] R. Duangsoithong and T. Windeatt, "Relevance and redundancy analysis for ensemble classifiers," *Mach. Learn. Data Min. Pattern Recognit.*, pp. 206–220, 2009.

[62] M. A. Hall, "Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 359–366.

[63] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.

[64] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," *Knowl.-Based Syst.*, vol. 86, pp. 33–45, 2015.

[65] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, 2016.

[66] S. Alelyani, H. Liu, and L. Wang, "The Effect of the Characteristics of the Dataset on the Selection Stability," in *Proceedings 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, 2011, pp. 970–977.

[67] D. Dernoncourt, B. Hanczar, and J.-D. Zucker, "Analysis of feature selection stability on high dimension and small sample data," *Comput. Stat. Data Anal.*, vol. 71, pp. 681–693, 2014.

[68] A.-C. Haury, P. Gestraud, and J.-P. Vert, "The Influence of Feature Selection Methods on Accuracy, Stability and Interpretability of Molecular Signatures," *PloS One*, vol. 6, no. 12, pp. 1–12, 2011.

[69] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and Communications Technologies for Sustainable Development Goals: State-of-the-Art, Needs and Perspectives," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 3, pp. 2389–2406, 2018.

[70] J. Wu, S. Guo, J. Li, and D. Zeng, “Big data meet green challenges: Greening big data,” *IEEE Syst. J.*, vol. 10, no. 3, pp. 873–887, 2016.

[71] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, and Y. Yang, “Big Data Meet Cyber-Physical Systems: A Panoramic Survey,” *IEEE Access*, vol. 6, pp. 73603–73636, 2018.

[72] J. Wu, S. Guo, J. Li, and D. Zeng, “Big data meet green challenges: big data toward green applications,” *IEEE Syst. J.*, vol. 10, no. 3, pp. 888–900, 2016.

[73] “Pima Indians Diabetes Data Set.” [Online]. Available: https://archive.ics.uci.edu/ml/support/pima+indians+diabetes.

[74] “Spambase Data Set.” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase.

[75] “UCI Machine Learning Repository: Data Sets.” [Online]. Available: https://archive.ics.uci.edu/ml/datasets.html.

[76] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar, “Streamwise feature selection,” *J. Mach. Learn. Res.*, vol. 7, no. Sep, pp. 1861–1885, 2006.

[77] J. Wang, P. Zhao, S. C. Hoi, and R. Jin, “Online feature selection and its applications,” *IEEE Trans. Knowl. Data Eng.*, pp. 1–14, 2013.

[78] Z. Zhao, L. Wang, H. Liu, and J. Ye, “On similarity preserving feature selection,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, 2013.

[79] Y. Zhai, M. Tan, I. W. Tsang, and Y.-S. Ong, “Discovering Support and Affiliated Features from Very High Dimensions,” in *Proceedings of the 29 th International Conference on Machine Learning*, 2012.

[80] “The Spider.” [Online]. Available: http://people.kyb.tuebingen.mpg.de/spider/.

[81] J. Liang and Z. Shi, “The information entropy, rough entropy and knowledge granulation in rough set theory,” *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 12, no. 01, pp. 37–46, 2004.

[82] Y. Qian and J. Liang, “Combination entropy and combination granulation in rough set theory,” *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 16, no. 02, pp. 179–193, 2008.

[83] C. E. Shannon, “A mathematical theory of communication,” *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.

[84] H. Yang and J. Moody, “Feature selection based on joint mutual information,” in *Proceedings of international ICSC symposium on advances in intelligent data analysis*, 1999, pp. 22–25.

[85] G. H. John and P. Langley, “Estimating continuous distributions in Bayesian classifiers,” in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, 1995, pp. 338–345.

[86] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.

[87] J. R. Quinlan, “Bagging, Boosting, and C4.S,” in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, 1996, vol. 1, pp. 725–730.

[88] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[89] F. Cismondi *et al.*, “Reducing unnecessary lab testing in the ICU with artificial intelligence,” *Int. J. Med. Inf.*, vol. 82, no. 5, pp. 345–358, 2013.

[90] J. Iavindrasana, G. Cohen, A. Depeursinge, H. Müller, R. Meyer, and A. Geissbuhler, “Clinical data mining: a review,” *Yearb Med Inf.*, pp. 121–133, 2009.

[91] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*, 3rd ed. Elsevier, 2011.

[92] Q. Liu, B. Ribeiro, A. H. Sung, and D. Suryakumar, “Mining the Big Data: The Critical Feature Dimension Problem,” in *Proceedings 2014 IIAI 3rd International Conference on Advanced Applied Informatics*, 2014, pp. 499–504.

[93] “MIMIC II Database.” [Online]. Available: https://mimic.physionet.org/database.html. [Accessed: 20-Aug-2015].

[94] J. Lee, S. Govindan, L. Celi, K. Khabbaz, and B. Subramaniam, “Customized prediction of short length of stay following elective cardiac surgery in elderly patients using a genetic algorithm,” *World J Cardiovasc Surg*, vol. 3, no. 5, pp. 163–170, Sep. 2013.

[95] L. Li-wei, S. Mohammed, T. Daniel, M. Roger, and M. Atul, “Methods of blood pressure measurement in the ICU,” *Crit Care Med*, vol. 41, no. 1, pp. 34–40, 2013.

[96] L. Li-wei, L. William, S. Mohammed, and M. Roger, “Latent topic discovery of clinical concepts from hospital discharge summaries of a heterogeneous patient cohort,” in *Proceedings of the 36th International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 1773–1776.

[97] “Laboratory Test Reference Ranges | Calgary Laboratory Services.” [Online]. Available: https://www.calgarylabservices.com/lab-services-guide/lab-reference-ranges/. [Accessed: 03-Sep-2015].

[98] “Feature Selection Package Documentation.” [Online]. Available: http://featureselection.asu.edu/documentation/infogain.htm. [Accessed: 04-Sep-2015].

[99] J. Platt, “Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines,” *Adv. Kernel Methods—support Vector Learn.*, vol. 3, 1999.

[100] “LOINC Codes - Mayo Medical Laboratories.” [Online]. Available: http://www.mayomedicallaboratories.com/test-catalog/appendix/loinc-codes.html. [Accessed: 10-Sep-2015].

[101] “ARUP Laboratories: A National Reference Laboratory.” [Online]. Available: http://www.aruplab.com/. [Accessed: 10-Sep-2015].

[102] “UCSF Departments of Pathology and Laboratory Medicine | Lab Manual | Laboratory Test Database | Activated Partial Thromboplastin Time.” [Online]. Available: http://labmed.ucsf.edu/labmanual/db/data/tests/802.html. [Accessed: 10-Sep-2015].

[103] “2345-7.” [Online]. Available: http://s.details.loinc.org/LOINC/2345-7.html?sections=Comprehensive. [Accessed: 10-Sep-2015].

[104] X. Wu et al., “Top 10 algorithms in data mining,” *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[105] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics New York, 2001.

List of Publications

Accepted (With Revision)

- **Noura AlNuaimi** and Mohammad Mehedy Masud, “Online Streaming Feature Grouping and Selection for Big Data Classification,” *WIREs Data Mining and Knowledge Discovery,* 2019. (Accepted with revision April 6, 2019).

Accepted *(In Press)*

- **N. AlNuaimi**, M. M. Masud, M. A. Serhani, and N. Zaki, “Streaming Feature Selection Algorithms for Big Data: A Survey,” *Journal of Applied Computing and Informatics*, 2019.

Published

- **N. Al Nuaimi** and M. M. Masud, “Toward Optimal Streaming Feature Selection,” in *Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on*, 2017, pp. 775–782.

- **N. AlNuaimi**, M. M. Masud, and F. Mohammed, “Examining the effect of feature selection on improving patient deterioration prediction,” *Int. Journal of Data Mining and Knowledge Management Process IJDKP Vol*, vol. 5, no. 6, pp. 13–33, 2015.

- **N. AlNuaimi**, M. M. Masud, and F. Mohammed, “ICU Patient Deterioration prediction: a Data-Mining Approach,” *CoRR*, vol. abs/1511.06910, 2015.

- **N. Al Nuaimi**, A. AlShamsi, N. Mohamed, and J. Al-Jaroodi, “E-Health cloud implementation issues and efforts,” *International Conference on Industrial Engineering and Operations Management (IEOM)*, 2015, pp. 1–10.

- **N. Al Nuaimi**, “Data mining approaches for predicting demand for healthcare services in Abu Dhabi,” *Innovations in Information Technology (Innovations), 2014 10th International Conference on*, 2014, pp. 42–47.