Iraqi Journal of Statistical Sciences

https://stats.mosuljournals.com/

[Mining Streaming Database: A Review

Azhar Muhammed Salih[1] and Ammar Thaher Yaseen Al Abd Alazeez [2]

[1,2]Department of Computer Science, College of Computer Science and Mathematics University of Mosul, Mosul, Iraq

Article information

Article history:
Received :May 7,2024
Accepted: June 21,2024
Available online : December 1,2024

Keywords:
Flexible parametric survival model, Additive hazard model, Cox model, Time-varying covariate, Tuberculosis

Correspondence:
Azhar Muhammed Salih
azhar.22csp24@student.uomosul.edu.iq

Abstract

Background: Tuberculosis (TB) is a globally deadly infectious disease responsible for 10 million new cases and 1.5 million deaths annually. Shorter TB treatment regimens show promise in reducing this problem, but there is an improved treatment success rate in South Africa, while retreatment cases remain a concern. An important feature of time-to-event modelling is its ability to consider transition probabilities of heterogeneous subgroups with different risk profiles. Survival analysis is generally performed to accurately estimate the transition probabilities associated with the risk profiles. This study explored the application of a flexible parametric survival model for analysing censored time-to-event data among TB patients.

Methods: The data were obtained from East London Central Clinic-TB unit, Eastern Cape, South Africa. In total, 174 patients were included in the analysis. The goodness of fit of the models was explored using AIC. We estimated the hazard ratios and baseline cumulative hazards of our model, which are necessary to calculate individual transition probabilities, and compared the model with the Cox model and additive hazard model to determine the survival predictions of TB patients.

Result: The flexible parametric survival model produced hazard ratio and baseline cumulative hazard estimates that were similar to those obtained using the Cox proportional hazards model. The analysis revealed that sex (HR=0.49, 95% CI: 0.38, 0.62), antiretroviral therapy, ART (HR=0.53, 95% CI: 0.34, 0.78), and diabetes (HR=0.58, 95% CI: 0.41, 0.78) were all statistically significant factors associated with improved treatment survival in tuberculosis patients.

Conclusion: Flexible parametric survival models are a powerful tool for modelling time-to-event data and individual transition probabilities. It is of great importance to fit models by modelling the baseline, which makes it easier to make different types of predictions and allows for non-proportional hazards since it is an interaction.

DOI 10.33899/iqjoss.2024.185248, ©Authors, 2024, College of Computer Science and Mathematics University of Mosul.

1.Introduction

A Database DB is a group of related data, used by a DBMS. Database system is a means of storing and managing internally related data instead of the outdated, straightforward file system, which only allowed access to the data stored there through designated methods of storage and retrieval, database systems allow for the management and archiving of internally related data [1]. NoSQL database is short for not only SQL, but it is also known as a non-relational database (that is, it does not have any fixed relationships or schemas) or a distributed database (because the data in a NoSQL database management system DBMS can be divided into servers). It's a database management system which does not require the data to display a pre-defined schema. It allows a variety of data types to be stored. This implies that the data can be arranged in a variety of ways based on the database's intended use [2].

The number of connected devices, such as sensors, cameras, and radars that communicate through local networks, Networked IoT, and social networks, leads to the generation of a significant amount of streaming data. Data from these sources are often generated continuously, and are called data streams. In a streaming data environment, time is a very important constraint [3]; the data stream always changes over time. it is temporary and there is no upper limit to the data. Data is produced at high-speed streaming data environment. Considering all these parameters the data stream processing has many limitations, which must be processed in a limited amount of time as well as limited space usage[4]. An incoming series of data (also known as a data stream) has been designed to be collected and processed by a streaming database. instantaneous data in real time following its creation. Instead of referring to a distinct class of database management systems, this term encompasses a variety of databases that process live streaming data. Consisting of time-series databases, NewSQL databases, NoSQL databases, and in-memory data grids [5].

Data streaming mining is a method to extract useful knowledge and obtain relevant information from data. Data classification is one of the data mining techniques that has shown its power in predicting large amounts of data. There are different classification methods, traditional ones such as decision trees, rule-based methods, neural networks, and the simple Naive Bayes method, which have shown tremendous performance in various applications. These techniques are often used in streaming data mining where stream classifiers operate on data in a single pass because they are time-constrained. The accuracy of data classifiers depends on class label prediction, and in stream data identification of proper label is very difficult [6]. This research paper is organized as follows, Section 2 presents the NoSQL database and its features, characteristics and types. Section 3 presents data stream data, its characteristics, processing methods, data stream processing tools, and the stream database. Section 4 presents data stream mining, its characteristics, limitations and challenges facing stream data classification, and the algorithms used in data stream classification.

2. Related Works

Researchers [7] proposed a data streaming method to predict solar radiation in real time with variable climate conditions and cloud coverage. Our method works within an asynchronous two-pipeline framework and using deep learning models, in order to evaluate four deep learning architectures, multilayer perceptron (MLP), long. -short term memory network (LSTM), convolutional network (CNN), and Transformer network, where (MLP) and (CNN) achieved the best accuracy with a high ability to adapt to evolving data. Researchers [8] proposed a new nonlinear feature selection method targeting multi-class classification problems in the framework of support vector machines. The proposed method is achieved using a multi-class support vector machine with a fast version of recursive feature removal. The proposed method selects features that perform well for all classes. The classifier in question simultaneously builds multiple decision functions that separate each class from the others. Researchers [9] proposed a data streaming method to predict solar radiation in real time with variable climate conditions and cloud coverage. Our method works within an asynchronous two-pipeline framework and using deep learning models, in order to evaluate four multilayer perceptron (MLP), long-short, deep learning architectures. term memory network (LSTM), convolutional network (CNN), and Transformer network, where (MLP) and (CNN) achieved the best accuracy with a high ability to adapt to evolving data. Researchers[10] presented a study on the impact of assessment scores and online activity data in a learning management system(LMS) on students' academic performance. Based on one of the commonly used data mining techniques for forecasting,. Five classification algorithms were applied, decision tree, random forest, sequential minimal optimization, multilayer perceptron, and logistic regression, and the results showed that assessment scores are the most important attributes that affect students' academic performance. Prediction models that included assessment scores alone or in combination with activity data performed better than models based on activity data alone. The random forest algorithm performed well for predicting student academic performance, followed by decision tree.

3.NoSQL Database

The term NoSQL was first used in 1998 in databases. Its importance has become increasing during the twenty-first century, particularly with considering quickly of the Internet. Due to the increasing acceptance of Web services, traditional databases are being used more frequently on a web scale, which has resulted in the creation of massive amounts of data. New data management systems were therefore required in order to handle enormous amounts of streaming data from various sources [11]. NoSQL databases are distributed database systems, which do not require structured data, are designed to scale horizontally and may be open source. NoSQL databases do not fully provide the standard ACID (Atomicity, Consistency, Isolation, and Durability) properties that are provide relational databases. NoSQL databases are characterized by the ability to manage, more flexibly store and index massive data sets whereas accepting numerous concurrent user requests. NoSQL databases are now being widely used in various application fields, including manufacturing, healthcare, and social media.

NoSQL was mainly used to process the drawbacks of relational database systems RDMBS in web applications; the drawbacks ware related to performance and scalability [12].

3.1 NoSQL Database Characteristics [13]:

1. Non-Relational: Relational database models are not used by NoSQL databases, and they are also not compatible with SQL join operations.
2. Distributed: NoSQL databases typically store data across multiple servers and geographical locations.
3. Open-Source: The majority of NoSQL databases are free to download and open source.
4. Horizontally Scalable: Reduce or increase the total number of servers to match the NoSQL database's processing capacity for data.
5. Schema-Free: NoSQL databases do not require the definition of database schema prior to data inserting, in contrast to RDBs. As a result, NoSQL databases allow for flexible data addition.
6. Simple API: The NoSQL database provides APIs for network delivery, data collection, etc. for programmers to use, so that programmers do not need to design additional programs to make writing programs easier.
7. Its commitment to the characteristics of BASE: It is an abbreviation for Basically Available Soft-State and Eventual Consistency, and the meanings are described as follows:

A. Basically Available: The database system can execute and always provide services. Some parts of the DB system may have partial failures and the rest of the database system can continuously operate. Some NoSQL DBs typically keep several copies of specific data on different servers, which allows the database system to respond to all queries even if few of the servers fail.

B. Soft-State: The database system does not require a state of strong consistency. Strong consistency means that no matter which replication of a certain data is updated, all later reading operations of the data must be able to obtain the latest information.

C. Eventual Consistency: The database system needs to meet the consistency requirement after a certain time. Sometimes the database may be in an inconsistent state. For example, some NoSQL DBs keep multiple copies of certain data on multiple servers. However, these copies may be inconsistent in a short time, which may happen when a copy of the data is updated while the other copies continue to have data from the old version. Eventually, the replication mechanism in the NoSQL database system will update all replicas to be consistent.

3.2 NoSQL Database Properties

Brewer's CAP Theorem demonstrates that distributed systems are only able to have two of the following characteristics at once [14]:

1. Consistency: shows that each read retrieves the most lately write.
2. Availability: suggests that writing and reading are always successful. Stated alternatively, every non-failing node will provide a response within an appropriate amount of time.
3. Partition Tolerance: This indicates that the system will keep working regardless of a system malfunction or data loss occurs.

In most cases, NoSQL databases prioritize high availability and partitioning. Which leads to loss of data consistency. Which in turn leads to the creation of systems known BASE mentioned previously.

3.3 NoSQL Database Types

NoSQL databases are those that offer an alternative mechanism for storing data than SQL tables. Four different kinds of NoSQL databases exist:

1. Key-Value Database

A key-value database stores data in pairs, each of which consists of a key and a value. The arrangement of information (values) that the database can hold is the focus of the key, and the value corresponds to the key. This structure is quick to query and simple.. The advantages of Key-Value are size and scalability [12,15]. An example of this type is the Riak database.

2. Document Oriented Database

A document store or document-oriented database. In terms of structure, this model is comparable to Key value-based models; however, the value documents are semantic, and is stored in JSO or XML format. This type is schema less, which helps in heterogeneous semi-structured data storage, and high flexibility. A reference key in a document-oriented database is similar to a foreign key in a relational database. The keys are unique and are associated with each document collection [12,15]. An example of this type is the MongoDb.

3. Column Database

A column database stores data by columns instead of rows, and is stored separately for each column feature which uses the table as a data model, making it ideal for storing data and processing big data. Columnar databases are widely used in data warehouse environments. Column database is known for its high performance and can support complex data types, unstructured text and graphics such as jpeg, gif, bmp, etc. [15]. Cassandra is an example of such a database.

4. Graph Database

Graphs can be used to represent data and can be more relational than a relational database. Graph databases are employed when the goal is to extract intricate relationships from massive data sets. These databases are extensively used by social media networks. Graphs are frequently utilised in link structures on websites. [12,15]. An example of this type is the OrientDb database.

4. Data Stream

Technology has been more prevalent in many societal sectors recently, including banking, transportation, and healthcare. More sensors and systems that continuously produce enormous amounts of data as data streams are progressively added as part of this digital revolution. These systems and sensors include, for instance, weather forecasting, drones, and cell phones. Massive data streams are produced by these devices, and it is anticipated that these data streams will increase soon [16]. A data sequence with a timestamp is called a data stream. Data is variable; it comes in at a high rate and speed, is generated by various applications, and may change over time. The massive volume of data in the data stream is processed in real time, at a high rate of speed, and in an infinite size with no set length or at least unknown length [17].

4.1 Data Stream Characteristics

Streaming data has the following properties [18]

1. Timestamp: Each item in streaming data has a timestamp, which is time sensitive with decreasing importance after a period of time.
2. Continuous: The data stream has no beginning or end That is, data is collected continuously.
3. Unique: Repeated transmission of data streams represents a real challenge due to time sensitivity. Accurate processing of data in real time is crucial. This means that it processes each element in streaming data at least once.
4. Heterogeneous: Data arrives from different sources, so the data is different.
5. Very large data: The size of the data may be huge, so it is difficult to store all the stream data in memory.

4.2 Data Stream Processing

It is the processing unit responsible for standardizing formats to analyze the collected data and then extract the relevant information (i.e. converting the received data from its raw form into usable and more readable information). It is also responsible for removing potentially incorrect data. This processing can be performed in two ways: Stream Processing and Batch Processing [19]. Table (1) shows the difference between the two methods.

Table (1): The difference between Stream Processing and Batch Processing [19]

Batch Processing	Stream Processing
processing is in the form of a large ntity of data in one batch.	is processed continuously during its uction
response time (minutes to hours)	response time (milliseconds)
data's size is known and specified	data's size is unknown and undefined
process it is without an internet nection and there is no real time	cessing is done online in real time
work is scheduled to be carried out	lementation of work is continuous
ed batch data	is variable and dynamic
w data generation	data generation

4.3 Streaming Database

is a kind of database created especially to handle massive volumes of real-time data. data in real time. Analytics and insights are possible in real-time thanks to streaming databases, which process data as soon as it is created. For latency-critical applications like real-time analytics, fraud detection, and network monitoring, streaming databases are perfect [20].

4.4. Streaming Database Processing Tools

1. Apache Spark

It is an open-source framework for real-time data entry, storing, processing, and analysis. It primarily uses small batch processing mode, in which events are handled according to predetermined time frames. It allows for the processing of multiple data types, expanding the capabilities of the MapReduce algorithm. Spark provides specialised libraries known as Spark Streaming to facilitate stream processing with short latency. For algorithms involving iterative processing, Spark is an excellent tool. To implement algorithms, a variety of programming languages can be used, including Java, Python, R, and Scala [21]. Data streams are managed by Spark Streaming as a sequence of small-batch jobs. hence achieving 100 ms end-to-end latencies. A Structured Streaming library based on the Spark SQL engine was introduced by Spark. The foundation of structured streaming is a brand-new, end-to-end latency-achieving low-latency processing mode called Continuous Processing.as little as one millisecond [22]. Figure (1) shows the Apache Spark architecture.

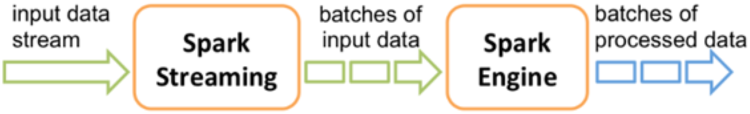

Figure (1): shows Apache Spark

2. Apache kafka

It is an open source software platform for processing data streams in real time. Designed specifically for Distributed Messaging System, it collects, delivers and processes large amounts of data with low latency and access. The Kafka architecture basically consists of three parts (see Figure (2)): The Producer, which collects data from different sources and then inserts it into the Topic, where the Topic describes the Data Stream Event and contains the Queue Structure. The third element of the Kafka architecture is the consumer, which retrieves topic elements [23].

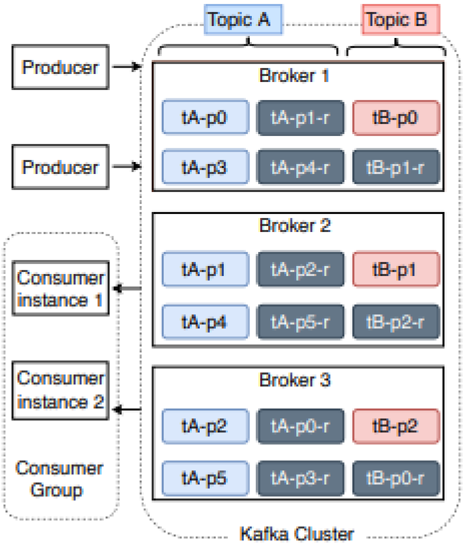

Figure (2): Apache Kafka architecture

3.Apache Strom

It is an open-source framework made for processing massive volumes of data with ease and speed while enabling in-depth analysis. Programmers can create real-time distributed processing systems with the aid of Apache Strom. Its

Application Programming Interface (APL) is straightforward. Apache Strom uses a series of tuples to represent streaming data. figure (3) shows the architecture of Apache Strom. Spout is the source of the data stream. A bolt processes any number of inputs streams and produces any number of new output streams. Topology is a Network that contains bolts and spouts as well as data stream between them [24].

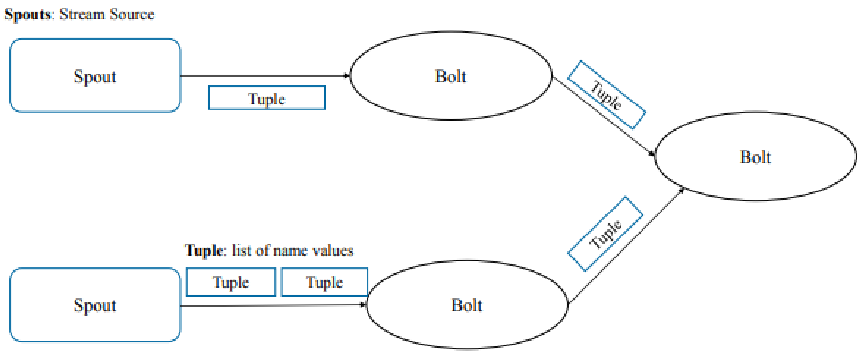

Figure (3): shows Apache Strom

3. Apache Flink

It is framework for distributed processing and open source. It is a platform designed to process sets of streaming data from different inputs. Scala and Java are the languages used in its development, which is done by the ASF. No latency, fault tolerance, and high throughput are offered by the framework. With its high-level libraries for Java, Python, Scala, and SQL, Apache Flink facilitates both batch and stream processing [25].

5. Streaming Database Mining

Numerous tasks, including frequency counts, time series analysis, clustering, and classification, are included in streaming data mining. Over the past ten years, the data stream classification task has been the most important and focused area of research among all these tasks. The traditional static data classification task differs slightly from data stream classification [26]. Table (2) shows a comparison between traditional data mining and data stream mining.

Table (2): Comparison between traditional and streaming data mining [27]

Advantage	Traditional data mining	Streaming data mining
Processing	Offline	Online
Storage	Possible	Impossible
Size	Specific	No Specific
Data generation	Static	Dynamic
Time	More time to access data	Just one pass
Data type	Homogeneous	Heterogeneous
Result	Accurate	Approximation

5.1 Characteristics of Mining Streaming Database

Data stream mining has several unique characteristics [27]:

1. Size : Such data cannot be stored for further mining due to the massive size of the data stream. Put another way, data mining is necessary to extract useful information from this limitless amount of data. The data stream contains a variety of data types, including text, pictures, and video files. As a result, the data stream mining is only partially saved.
2. Speed : The dynamic arrival of the data stream changes over time, and the high speed at which this data is generated places strict requirements on the effectiveness of data mining.
3. Multi-Dimensional : Enormous streams of data with a variety of data types are generated from sources that are widely dispersed. This requires the need for complex algorithms.. The enormous amount of streaming data that enables processing and calculation in a single pass presents significant challenges.

5.2 Constraints and Challenges Facing Classification Streaming Database

Provides research [28] mainly presents some common constraints for streaming data classification:

1. Single Pass : While data streams contain dynamic data instances that can only be read once, the static data sets used in traditional data mining can be read numerous times.
2. Real-Time Response : Real-time response is necessary for many real-world data stream applications, including credit card transactions and stock market forecasts. Lower time costs, or quicker response times, are essential for data processing and decision-making.
3. Bounded Memory : Many instances will be discarded due to limited memory storage and processing power, since the size of the data stream is open-ended with the arrival of data instances. As a result, only the data summary can be computed and stored to produce results that are acceptable approximations.
4. Drift Detection : The concept drift refers to the changing of data distributions in the Class Space, the Feature Space, or both spaces over time. Drift detection is useful for updating the built model for adapting to dynamic changes hidden in data streams.
5. Multi-Label Problems : In a data stream, each Instance is associated with one or more Labels. In this scene, such as a type of news stream, it may belong to more than one topic simultaneously. Multi-label data stream classification aims to classify each data instance into a predefined label or a label set.

5.3 Classification Streaming Database

The classification of data streams has become more significant in various real-time applications. A data stream classification method can be developed using a variety of methods [29]. Figure (4) shows the most important data stream mining techniques. Machine learning, statistics, databases, data warehouses, data retrieval, and decision support systems. Machine Learning is a rapidly growing system that research how computers learn or improve their performance based on data. Statistic This technique studies the analysis collection, interpretation and presentation of data. Data Mining has an inherent relationship with statistics. Database systems are primarily concerned with building, managing, and utilising databases for end users and organisations. Using data storage and data mining, modern database systems have integrated systematic data analysis capabilities on database data. Data from various sources and time periods are combined in a data warehouse. To create data cubes, it aggregates data into a multidimensional space. The science of finding information or documents within text or multimedia documents that may be available online is known as information retrieval. A computerised system called a decision support system aids in the making of decisions, assessments, and plans of action for businesses or organisations. To find answers to issues and make decisions, a decision support system gathers and evaluates a lot of data. [30].

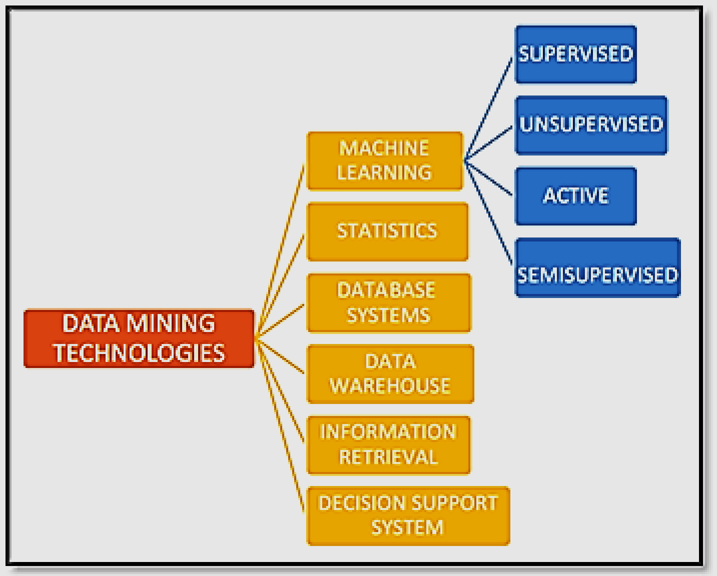

Figure (4): shows data mining techniques for data stream

5.3.1 Machine Learning Techniques for Streaming Database

The basic idea behind machine learning (ML), a branch of artificial intelligence, is that machines are capable of learning from data, identifying patterns, and making decisions without the need for human input. Data extraction is the goal of machine learning. Machine learning algorithms adapt to the style of data stream, which is dynamic, large, and continuous, as opposed to mining static data; instead, they develop an automated computational model and continuously improve its performance based on experience. supervised machine learning, unsupervised machine learning, semi-supervised machine learning, and reinforcement machine learning make up the four categories of machine learning. Classification and regression are two examples of supervised learning tasks. Supervised machine learning is one of the key areas of machine learning. One novel method is supervised machine learning [29]:

1.Machine learning is able to analyze massive quantity of data and identifying specific trends and patterns that may have gone undetected by humans.

2.It has the ability to learn on its own, make predictions, and make improvements to algorithms.

3.Machine learning algorithms gather experience, their accuracy and efficiency improve

4.Machine learning algorithms are capable of dealing with multidimensional and multivarious data and data in a dynamic or uncertain state.

5.3.2Classification Algorithms for Streaming Database

Classification algorithms in a streaming data environment are often designed based on some basic traditional machine learning algorithms [31], which may include the following algorithms:

1. Decision Tree

One simple technique for classification is a decision tree. Due to its low computational overhead, decision trees can be easily translated into highly accurate classification rules that are simple to comprehend and appropriate for high-dimensional data. Decision trees are therefore frequently used for streaming data and classification models [31]. Among the most popular algorithms in the academic world are decision trees. Tree models exhibit a high degree of interpretability and are free of distribution. These elements have played a significant role in its rising stature within the data mining community. For the data mining community, one of the most difficult problems is learning decision trees from data streams. One effective illustration of this is the Very Fast Decision Tree system, or VFDT. Repositioning leaves with decision nodes repeatedly teaches the very fast decision tree algorithm (VFDT). Sufficient information about attribute values is stored in each leaf. The statistics required by the evaluation function, which assesses the benefits of split tests based on attribute values, are considered sufficient. It moves from the root to the leaf of the tree, assessing [32]. This is the VFDT algorithm(1)

2. Naive Bayes

The Naive Bayes algorithm is a straightforward probability classifier that determines a set of probabilities by figuring out how frequently certain values occur and how they might combine in a given data set. Being a lossless classifier, the Naive Bayes algorithm generates a classifier that is functionally equivalent to the corresponding classifier that was trained on batch data. The foundation of the Naive Bayes algorithm is the Bayes theorem, a mathematical expression that expresses conditional probability. Taking into account the value of the class variable, this algorithm makes the assumption that all variables are independent. This conditional independence assumption is considered naive because it is rarely true in practical applications. In a range of controlled classification tasks, this algorithm has a tendency to pick things up quickly [33]. This is the Naive Bayes Algorithm (2).

```
Input :  S:Stream of examples
         X: Set of nominal Attributes
         Y:Y={$y_1$,……,$y_k$} Set of class labels
         H(.): Split evaluation function
         $N_m$in :Minimum number of example
         ʃ: is one minus the desired probability
         of choosing the correct attribute  at any node.
         T: Contant to solve ties
Output :  HT: is a Decision Tree
Begin
   Let HT← Empty leaf (Root) ;
   Foreach example ($x,y_k$) ϵ S do
       Traverse the tree HT from the root till a leaf I;
       If $y_k$ ==? Then
           Classify the example with the majority class in the leaf l;
       else
           Update sufficient statistics
           If Number of examples in l >$N_{min}$ then
               Compute $H_I(x_i)$ for all the attributes ;
               Let $X_a$ be the attribute with highest $H_I$ ;
               Let $X_b$ be the attribute with second highest $H_I$ ;
               Compute $\epsilon = \sqrt{\frac{Rln(2/ʃ)}{2n}}$ (Hoeffding bound)
               If ($H(X_a) - H(X_b) > \epsilon$ ) then
                   Replace I with a splitting test based on attribute $X_a$ ;
                   Add a new empty leaf for each branch of the split ;
               else
```

Algorithm(1): shows Very Fast Decision Tree Algorithm

```
function INITIALIZE (attributes,labels)
    (A,Y)← (attributes, labels)
    foreach class c  in C do
        N(c) =0
    n=0
    return
function TRAIN (X, y)
    Increase n by 1
    Increase number N(y) by 1
    Calculate probability P(y) =N(y)/n
    foreach attribute $a_i$ € A that holds the value of $x_i$ in x do
        Increase number  $N(x_i|y)$ by 1
        foreach class c in C do
            Calculate probability $P(x_i|c) = N(x_i|c) / N(c)$
    return
function TEST(X)
    foreach class c in C do
        Calculate likelihood $P(x|c) = \pi^{|x|}_{k=1} P(x_k|c). P(c)$
        Calculate relative – likelihood $Rel(c|x) = P(x| c) .P(c)$
    return the class with the maximum relative-likelihood
```

Algorithm (2) shows Naive Bayes Algorithm

3. Neural Network

Artificial intelligence that uses neural networks aims to mimic the functions of the human brain. When a neural network has a sizable database, it performs especially well in event prediction and pattern recognition [34]. Recently, promising results for text classification have been demonstrated by neural network-based applications. One of the key components of NLP, or natural language processing, is text classification. which has been used in a variety of fields, including information retrieval, document classification, sentiment analysis, and text classification. Neural network-based models are becoming more widespread. An artificial neural network with multiple layers of inputs and outputs is called deep learning. The RNN, or Recurrent Neural Network, is one of the deep learning techniques utilised [35]. The primary applications of a recurrent neural network are in sequential labelling and prediction. From sequential and time-series data, a recurrent neural network can learn features and long-term dependencies. Any dynamic system can be modelled by a properly trained RNN. One type of supervised machine learning model is the RNN. The input layer, the hidden recurrent layer, and the output layer are the three layers of a basic RNN network. Recurrent neural networks operate on the basis of reusing a layer's output by feeding it back into the input layer to aid in layer prediction.as depicted in Figure (5). [34].

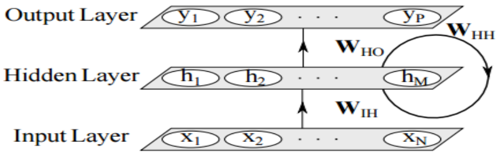

Figure (5) shows Recurrent Neural Network

N input units make up the input layer. A series of vectors over time T, such as {..., xt−1, xt, xt+1,...}, where Xt = (x1, x2,..., Xn), serve as the input for this layer. In a fully connected RNN, the hidden units in the hidden layer are connected to the input units. The hidden layer contains M hidden units, ht = (h1, h2,..., hM), where the connections are defined by the weight matrix WlH. These hidden units are connected to one another over time through recurrent connections, which may result in the initialization of the hidden units using small, non-Zero elements. can enhance the network's overall stability and performance. System memory is defined by the hidden layer.

$$ht = fH(ot) \quad (1)$$

where

$$\mathrm{ot} = \mathrm{WIHxt} + \mathrm{WHH\,ht} - 1 + \mathrm{bh} \quad (2)$$

fH()is the hidden layer activation function, b_h is the bias vector of hidden units. The hidden units are connected to the output layer through weighted W_{HO} connections. The output layer contains P units, where $y_t = (y_1, y_2, ..., y_P)$ that is calculated as follows:

$$\mathrm{yt} = \mathrm{fO}(\mathrm{WHOht} + \mathrm{bo}) \quad (3)$$

Where fo() is the activation function and b_o is the bias vector in the output layer, and since the input target pairs are sequential over time, the above steps are repeated over time [36].

5. Conclusion

This study reviewed the four types of NoSQL databases, along with their features and benefits. Key-Value, Document-Oriented, Graph, Column, and Database types, as well as how each should be used based on the needs of the system and the type of data being stored. The Key-Value database is used if the nature of storing the data is as a set of Principal value pairs. The Document database is used if the nature of storage is JSON objects. Graph databases are used when the goal is to capture complex in connections a huge volume of data. When data is stored by columns instead of rows in an application, it is recommended to use the Column database. The research paper also discussed data stream data, how to process it, and the tools used to process data stream. Research in the field of data stream mining is many and varied, especially classification algorithms. Each algorithm has its advantages and limitations, and given the research problem with the specific domain, the algorithm is chosen. Choosing the appropriate algorithm becomes a crucial task as this requires scientific background knowledge of the basic principles and methodological approaches in the research field. This survey aims to provide a coherent analysis of some data stream classification algorithms. Dynamic feature selection should attract more attention in high-dimensional data streams such as biological, chemical, textual, or space image applications. Furthermore, incorporating temporal ordering into software applications greatly helps in keeping track of developments in data stream mining. With this visualization should come a new tool so that users can easily and conveniently explore and understand the developments. There is another group of research directions, for example, measuring the degree of conceptual deviation, offloading data streams to cloud computing and multi-data stream.

References

1. Taipalus, T. (2023). Database management system performance comparisons: Asystematic literature review. *Journal of Systems and Software*, 111872,doi:https://doi.org/10.1016/j.jss.2023.11187.
2. Goltsis, A. (2022). A Performance Comparison of SQL and NoSQL Database Managem Sys tems for 5G Radio Base Station Configuration.
3. Hu, H. (2022). Solving the challenges of concept drift in data stream classification, doi: https://doi.org/10.18297/etd/3947 .
4. Samant, R. C., & Patil, S. H. (2022, May). A Systematic and Novel Ensemble Construction Method for Handling Data Stream Challenges. In *International Conference on Image Processing and Capsule Networks* (pp. 260-273). Cham: Springer International Publishing, doi: https://doi.org/10.1007/978-3-031-12413-6_20.
5. Hazelcast. (2024, January 9). Streaming Database: An Overview with Use Cases. https://hazelcast.com/glossary/streaming-database/
6. Samant, R. C., & Thakore, D. D. M. (2019). A rigorous review on an ensemble-based data .stream drift classification method. *Int. J. Comput. Sci. Eng*, *7*(5), 380-385, doi: https://doi.org/10.26438/ijcse/v7i5.380385.
7. Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M., & Riquelme, J. C. (2023). Short-term solar irradiance forecasting in streaming with deep learning. *Neurocomputing*, *546*, 126312.
8. Guo, Y., Zhang, Z., & Tang, F. (2021). Feature selection with kernelized multi-class support vector machine. *Pattern Recognition*, *117*, 107988
9. Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M., & Riquelme, J. C. (2023). Short-term solar irradiance forecasting in streaming with deep learning. *Neurocomputing*, *546*, 126312.

10. Alhassan, A., Zafar, B., & Mueen, A. (2020). Predict students' academic performance based on their assessment grades and online activity data. *International Journal of Advanced Computer Science and Applications*, *11*(4).
11. Meier,A., & Kaufmann, M. (2019). *SQL & NoSQL databases*. Berlin/Heidelberg, Germny:Springer Fachmedien Wiesbaden ., doi: https://doi.org/10.1007/978-3-658-24549-8_7.
12. Reddy, H. B. S., Reddy, R. R. S., Jonnalagadda, R., Singh, P., & Gogineni, A. (2022). Analy sis of the Unexplored Security Issues Common to All Types of NoSQL Databases. *Asian Journal of Research in Computer Science*, *14*(1), 1-12., doi: 10.9734/AJRCOS/2022/v14i130323.
13. Chen, J. K., & Lee, W. Z. (2019). An Introduction of NoSQL Databases based on their categoriesandapplicationindutries. *Algorithms*, *12*(5),106, doi: https://doi.org/10.3390/a12050106.
14. Chellappan, S., & Ganesan, D. (2019). *MongoDB Recipes: With Data Modeling and QuerBuilding Strategies*. A press., doi: https://doi.org/10.1007/978-1-4842-4891-1.
15. Palanisamy, S., & SuvithaVani, P. (2020, January). A survey on RDBMS and NoSQL Databases MySQL vs MongoDB. In *2020 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-7). IEEE., **doi:** 10.1109/ICCCI48352.2020.9104047.
16. Bahri, M., Bifet, A., Gama, J., Gomes, H. M., & Maniu, S. (2021). Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *11*(3), e1405., doi: https://doi.org/10.1002/widm.1405**.**
17. Agrahari, S., & Singh, A. K. (2022). Concept drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 9523-9540., doi: https://doi.org/10.1016/j.jksuci.2021.11.006.
18. Amazon Web Services ,20-12-2023,https://aws.amazon.com/what-is/streaming-data.
19. Benjelloun, S., El Aissi, M. E. M., Loukili, Y., Lakhrissi, Y., Ali, S. E. B., Chougrad, H., & El Boushaki, A. (2020, October). Big data processing: batch-based processing and stream-based processing. In *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)* (pp. 1-6).IEEE. **, doi:** 10.1109/ICDS50568.2020.9268684.
20. RisingWave ,20-12-2023, https://risingwave.com/blog/what-is-a-streaming-database/
21. Saini, H., Rathee, G., & Saini, D. K. (Eds.). (2020). *Large-scale Data Streaming, Processing, and Blockchain Security*. IGI Global.
22. Isah, H., Abughofa, T., Mahfuz, S., Ajerla, D., Zulkernine, F., & Khan, S. (2019). A survey of distributed data stream processing frameworks. *IEEE Access*, *7*, 154300-154316., **doi:** 10.1109/ACCESS.2019.2946884.
23. Chen, W., Milosevic, Z., Rabhi, F. A., & Berry, A. (2023). Real-Time Analytics: Concepts, Architectures and ML/AI Considerations. *IEEE Access*., **doi:** 10.1109/ACCESS.2023.3295694.
24. Kavitha, A. R., Simon, M. D., & Sumathy, G. (2023). Novel Fuzzy Entropy Based Leaky Shufflenet Content Based Video Retrival System. **, doi:** https://doi.org/10.21203/rs.3.rs-2424204/v1.
25. Biernat, N. A. (2020). *Scalability benchmarking of Apache Flink* (Doctoral dissertation, Kiel University).
26. Din, S. U., Shao, J., Kumar, J., Mawuli, C. B., Mahmud, S. H., Zhang, W., & Yang, Q. (2021). Data stream classification with novel class detection: a review, comparison and challenges. *Knowledge and Information Systems*, *63*, 2231-2276. , doi: https://doi.org/10.1007/s10115-021-01582-4.
27. Alothali, E., Alashwal, H., & Harous, S. (2019). Data stream mining techniques: a review. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *17*(2), 728-737. , doi: 10.12928/TELKOMNIKA.v17i2.11752.
28. Zheng, X., Li, P., Chu, Z., & Hu, X. (2019). A survey on multi-label data stream classification. IEEE Access, 8, 1249-1275**. , doi:** 10.1109/ACCESS.2019.2962059.
29. Arya, M. (2021). *Ensemble-based algorithm for efficient classification of real time data streams* (Doctoral dissertation, School of Computer Science, UPES, Dehradun).
30. Jiawei, H., Micheline, K., & Jian, P. (2011). Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems*, *5*(4), 83-124.
31. Guo, J., Wang, H., Li, X., & Zhang, L. (2021). Stream classification algorithm based on decision tree. *Mobile Information Systems*, *2021*, 1-11. , doi: https://doi.org/10.1155/2021/310305.
32. Gama, J. (2010). *Knowledge discovery from data streams*. CRC Press.
33. Mahdi, O. A. (2020). *Diversity Measures as New Concept Drift Detection Methods in Da ta Stream Mining* (Doctoral dissertation, La Trobe).

34. Islam, M., Chen, G., & Jin, S. (2019). An overview of neural network. *American Journal of Neural Networks and Applications*, *5*(1), 7-11. , doi: 10.11648/j.ajnna.20190501.12.
35. Sari, W. K., RINI, D. P., MALIK, R. F., & AZHAR, I. S. B. (2020, May). Sequential models for text classification using recurrent neural network. In *Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)* (pp. 333-340). Atlantis Press. , doi:_10.2991/aisr.k.200424.050.
36. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*. ,
doi:
https://doi.org/10.48550/arXiv.1801.01078.

تعدين قاعدة البيانات المتدفقة: مقال مراجعة

ازهار محمد صالح، عمار ظاهر ياسين ال عبدالعزيز

قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل، الموصل، العراق

azhar.22csp24@student.uomosul.edu.iq, *ammarthaher@uomosul.edu.iq*

الخلاصة: تطبق شركات الانترنت والمؤسسات الاخرى تقنية NoSQL بدلا من SQL التقليدية بسبب سرعتها في أدخال، استرجاع، ومعالجة كمية هائلة من البيانات. ان سهولة التصميم والقدرة على التوسع أفقيا وبساطة نموذج البيانات الخالي من المخططات جعلت الشركات تبدأ في أدراك ان قواعد البيانات NoSQL هي طريقة مميزة لإدارة الكميات الكبيرة من البيانات المنظمة وشبة المنظمة وغير المنظمة. يطلق على الكمية الهائلة من البيانات التي يتم انشاؤها من مصادر مختلفة وغير متجانسة وبشكل مستمر ولها اشكال واحجام مختلفة والتي تتطلب معالجة في الوقت الفعلي تسمى البيانات المتدفقة. قاعدة البيانات المتدفقة هي نوع مخزن للبيانات يهدف الى استقبال ومعالجة دفق مستمرة من البيانات الواردة. وبعبارة أخرى معالجة البيانات وتخزينها في الوقت الفعلي. تنقيب تدفق البيانات هي عملية استخراج وايجاد المعرفة والانماط من تدفقات البيانات التي يتم انشاؤها بشكل مستمر مع قلة الموارد. يسهل تنقيب دفق البيانات تحليل كمية هائلة من البيانات المتدفقة في الوقت الفعلي، ولهذا السبب يتعين على خوارزميات التنقيب المتدفقة معالجة كل مثيل بيانات في فترة زمنية محددة واجراء التنبؤات بها في اي وقت باستخدام كمية صغيرة فقط من الذاكرة كل ذلك أثناء العمل بشكل تدريجي وسريع. يعد الانحدار والتجميع والتصنيف من أكثر التقنيات شيوعاً المستخدمة في تنقيب تدفق البيانات. بالاضافة الى دراسة الصعوبات المرتبطة بتدفق البيانات والأدوات الازمة لمعالجتها وفقا لمتطلبات الوقت الفعلي. يهدف البحث في دراسة خوارزميات وتقنيات تصنيف قواعد البيانات المتدفقة،. تتمثل مساهمات هذه الدراسة في فهم قواعد البيانات NoSQL، والمعالجة والتنفيذ عبر الإنترنت، وتدفقات البيانات واختيار الأدوات الاكثر فعالية لاتخاذ قرارات التصنيف.

الكلمات المفتاحية: قواعد البيانات NoSQL، قواعد البيانات المتدفقة، تنقيب البيانات، التصنيف