David C. Wyld
Natarajan Meghanathan (Eds)

Computer Science & Information Technology

4th International Conference on Artificial Intelligence and Applications (AI 2018), October 27 ~ 28, 2018, Dubai, UAE

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Natarajan Meghanathan,
Jackson State University, USA
E-mail: nmeghanathan@jsums.edu

ISSN: 2231 - 5403
ISBN: 978-1-921987-92-2
DOI : 10.5121/csit.2018.81401 - 10.5121/csit.2018.81405

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 4th International Conference on Artificial Intelligence and Applications (AI 2018) was held in Dubai, UAE during October 27 ~ 28, 2018, 2018. The 4th International Conference on Computer Science and Information Technology (CSTY 2018) and The 4th International Conference on Networks, Mobile Communications and Telematics (NMOCT 2018) was collocated with The 4th International Conference on Artificial Intelligence and Applications (AI 2018). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The AI-2018, CSTY-2018, NMOCT-2018 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, AI-2018, CSTY-2018, NMOCT-2018 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the AI-2018, CSTY-2018, NMOCT-2018.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld
Natarajan Meghanathan

Organization

General Chair

David C. Wyld	Southeastern Louisisna University, USA
Jan Zizka	Mendel University in Brno, Czech Republic

Program Committee Members

Abdelhafid ZEROUAL	University of Artois, France
Abdelmajid Hajami	FST Settat, Morocco
Abraham Sanchez Lopez	Autonomous University of Puebla, Mexico
Adnan Albar	King AbdulAziz University, Saudi Arabia
Afshari	Islamic Azad University, Iran
Ahmed Korichi	University of Ouargla, Algeria
Alaa Hussein Al-hamami	Amman Arab University, Jordan
Amir Masoud Rahmani	Islamic Azad University, Iran
Annamalai	Prairie View A&M University, USA
Ayush Singhal	Contata Solutions, USA
Azeddine Chikh	University of Tlemcen, Algeria
Babar Shah	Zayed University, UAE
ChanChristine	University of regina, Canada
Claudio Gallicchio	University of Pisa, Italy.
Dac-Nhuong Le	Haiphong University, Vietnam
Daniel Ekpenyong Asuquo	University of Uyo, Nigeria
Dipak Kumar Jana	Haldia Institute of Technology, West Bengal
Dongping Tian	Baoji University of Arts and Sciences, China
Emad Awada	Applied Science University, Jordan
Emad Eldin Mohamed	Canadian University Dubai, UAE
Farhad Soleimanian Gharehchopogh	Islamic Azad University, Iran
Farida Bouarab-Dahmani	Mouloud Mammeri University of Tizi-Ouzou, Algeria.
Farzad Kiani	Istanbul Sabahattin Zaim University, Turkey
Farzin Piltan	University of Ulsan, Korea.
Fatma Outay	Zayed University DXB, UAE
Fernando Zacarias Flores	Universidad Autonoma de Puebla, Mexico
Gammoudi Aymen	University of Tunis, Tunisia
Hamid Ali Abed AL-Asadi	Basra University, Iraq
Hamzeh Khalili	Universitat Politecnica de Catalunya (UPC), Spain
Hanan Salam	University of Pierre and Marie Curie, France
Hao-En Chueh	Yuanpei University, Taiwan, Republic of China
Hari Krishna Garg	National University of Singapore, Singapore
Heldon Jose	Professor of Integrated Faculties of Patos, Brazil
Hossein Jadidoleslamy	The University of Zabol, Iran
Irina Perfilieva	University of Ostrava, Czech Republic
Isa Maleki	Islamic Azad University, Iran
Javid Taheri	Karlstad University,Sweden

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Networks & Communications Community (NCC)

Soft Computing Community (SCC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

4th International Conference on Artificial Intelligence and Applications (AI 2018)

4th International Conference on Computer Science and Information Technology (CSTY 2018)

4th International Conference on Networks, Mobile Communications and Telematics (NMOCT 2018)

EPHEMERAL PARTIALLY REPLICATED DATABASES

Ritesh Agrawal[1] and Peter I. Frazier[1,2]

[1]Uber Technologies, Inc., 555 Market St., San Francisco, CA, USA
[2]School of ORIE, Cornell University, 232 Rhodes Hall, Ithaca, NY, USA

ABSTRACT

In today's analytics-driven world, fully replicated isolated databases provide much-needed database availability and compute scalability but at the cost of storage scalability, an issue that is addressed by partially replicated isolated databases. However, a partially replicated database that is optimal at the time of design is soon made inefficient by changing business needs, products & services it offers, datasets and query workloads. To this address this issue, we introduce the notion of migration cost as an additional factor that influences the design of a partially replicated databases. In this paper, we formalize the notion of migration cost and present a new cost-based objective function to partition and allocate data elements across available databases. Further, we discuss its implementation in the context of Uber and demonstrate its effectiveness based on a 10-week simulation study.

KEYWORDS

Databases, Partially Replicated Databases, Mixed Integer Linear Optimization

1. INTRODUCTION

In today's data-driven economy, the design of a holistic database solution is governed by three factors: database availability[2, 7], compute scalability [5, 10]and storage scalability. Database availability is concerned with the up-time of the database and its ability to execute requests. Compute scalability is concerned with the query throughput of the database. Lastly, storage scalability is concerned with the amount of data that can be managed by the database. Different approaches exist to maximize the three different factors. One of the approaches to maximize database availability and compute scalability is to have a multiple fully replicated isolated databases. Fully replicated isolated databases consists of multiple isolated databases with the same data copied across all the databases . Further, the connection between the client applications and the databases is often abstracted using a thin service layer. Such a database design helps achieve a high degree of database availability by routing queries to operational databases and compute scalability by distributing the query load across all the operational databases[13].

Nevertheless, the compute scalability achieved using the fully replicated databases doesn't grow linearly with the number of database replications. Each new replication of the data requires additional copying of data from one database to another, and thereby significant amount of resources are wasted on reading and writing data[11]. Furthermore, while the fully replicated database system helps achieve a high degree of database availability and compute scalability, it fails to provide storage scalability. At Uber, achieving a high degree of storage scalability along

David C. Wyld et al. (Eds) : AI, CSTY, NMOCT - 2018
pp. 01–13, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81401

with database availability and compute scalability is important for multiple reasons. First, we rely on Vertica, a fast analytic engine, to munge over petabytes of data and provide critical business insights as quickly as possible; the mean average time to complete a query is about 17 seconds. Further, we have multiple fully replicated Vertica database cluster to maximize database availability and to handle millions of queries on a daily basis. However, Vertica's licensing fee is tied to the amount of data stored. As a result, each additional replication significantly increases the licensing cost. In particular, since the number of replications is governed by the compute scalability requirement rather than the database availability, there is a significant amount of storage, and licensing fee is wasted due to additional replications of all the data elements in the case of a fully replicated isolated databases. The other reason to maximize storage utilization is that Vertica operates at its peak performance on a limited amount of data. Full replication thereby limits our ability to maximize the amount of data that can be quickly processed across all the databases. Apart from storage scalability, a fully replicated isolated database system suffer from other challenges as well such as data consistency management[9], non-linear computational growth as each additional replication requires additional computational resource for writing and managing data[12].

A natural solution to achieve database availability, compute scalability and storage scalability is a partially replicated databases system. As shown in Figure 1, a partially replicated databases optimize for storage by copying different overlapping subsets of datasets to different databases. There are many different proposed approaches for determining an optimal partially replicated database configuration, but one challenge that remains unattended is the ephemeral nature of partially replicated isolated databases. Different business units grow at different rates and thereby associated data elements. As a result, partial database configurations eventually become imbalanced in terms of data distributions across databases. Furthermore, as business evolve so the analytical requirements. As a result, the query patterns might change over time, causing imbalances in the distribution of the query load. As a result, a partial database configuration eventually becomes sub-optimal and thereby requires rebalancing of partial databases by moving data elements from databases to another. This migration of data elements is termed here as "migration cost."

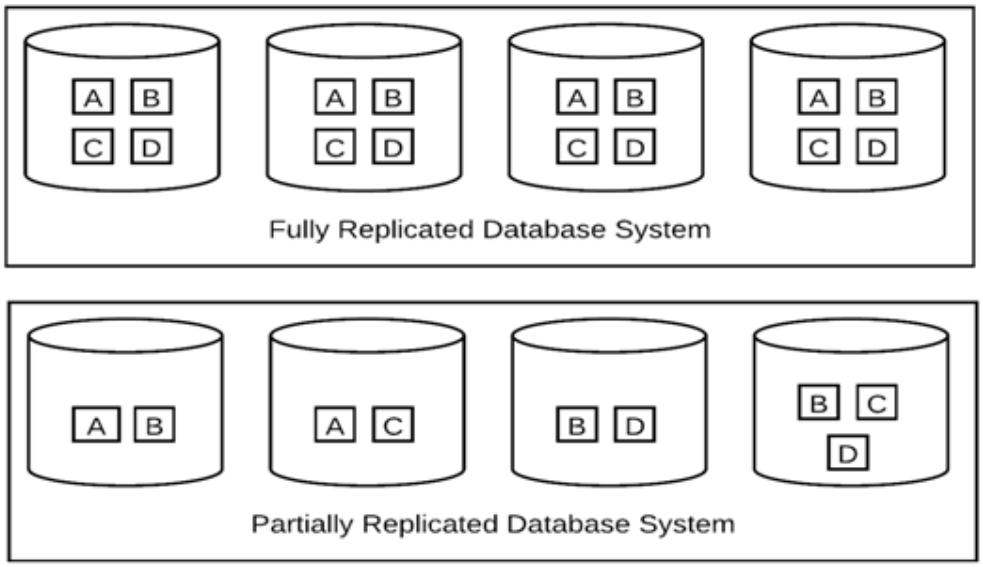


Figure 1. Sample Configuration of a fully replicated isolated databases and a partially replicated isolated databases. All data elements are replicated across all the databases in the case of a fully replicated

databases. In the case of a partially replicated isolated databases, different datasets contains different overlapping sets of data elements.

In this paper, we present a migration cost based formalization of partially replicated isolated databases and discuss its implementation in the context of Uber. In Section 2, we discuss prior approaches for designing partially replicated databases. In section 3, we discuss our formalization of the problem that explicitly incorporates migration cost for determining the optimal placement of data elements across available databases. In Section 4, we present the implementation details of our ephemeral partially replicated databases at Uber and the savings achieved as compared to the fully replicated databases. Lastly, in Section 5, we present our conclusion. Note that the scope of this paper is limited to isolated databases i.e. to complete a given query, all the required data elements should be part of a single database.

2. RELATED WORK

A partially replicated database system is described here as a system of multiple isolated databases with overlapping sets of data elements. Since these databases are isolated, all the data elements necessary to satisfy a query are required to be available on a single database. Designing a partially replicated database system is usually a two-step process involving partitioning and allocation[11].

Partitioning is concerned with generating partially overlapping sets of data elements. Previous efforts on partitioning mainly focused on "data marts", partitioning data elements to address specific problems and is often organized around various teams or products [3]. A more data-driven approach to partition is presented in [1]. In general, the data-driven approach uses historical query patterns to identify strongly correlated sets of data elements. Depending on how data elements are defined, these data-driven approaches can generate horizontal (dissecting data by rows) [4]or vertical (dissecting data by columns/attributes) partitioning of the data[1]. In contrast to partitioning, the allocation problem is concerned with assigning partitions of data elements to available databases and often studied as an optimization problem.

In [11], partitioning and allocation are considered as a single problem and proposed a data-driven approach to generate a partially replicated database configuration. The objective of their data-driven solution is to balance storage requirement and query load across various isolated databases. Queries are grouped into various classes based on the different sets of data elements required to satisfy a query and are further assigned compute cost. Then the objective for generating partially replicated database configuration is to assign these different classes of queries to different databases to balance data size and query load across all the databases.

One important aspect missing in previous approaches is the lack of consideration to the ephemeral nature of partially replicated database systems. Different product or services offered by a business grow at a different rate and thereby associated data size. Further, to maintain the competitive edge, business constantly add new services and products as well deprecate some. As a result, data elements and associated business queries continuously change. As a result, partially replicated databases often needs to be re-configured to adapt to the changing business and uneven workload. The cost of transitioning from one state of a partially replicated database to another is termed here as the migration cost. A high migration cost negatively hurts database availability and compute scalability as a significant amount of computing resources are spent on migrating data elements from one database to another. In this paper, we extend the idea presented in the above approaches and especially in [11] to explicitly account for the migration cost.

3. MIGRATION AWARE PARTIALLY REPLICATED DATABASES

Four factors influence the design of partially replicated isolated databases: database availability, compute scalability, storage scalability and migration cost. We approach the design of the partially replicated isolated databases as a single optimization problem that explicitly models the influence of all the four factors. Similar to[6, 11], our approach combines partitioning and

allocation steps in a single optimization problem. However, we first introduce key elements of our formalization in Section 3.1. Thereby, in Section 3.2, we introduce cost functions associated with each of the four factors and combine them into a single objective function.

3.1. Notations

Below are the various notations used while constructing our objective function.

1. Database (N): Let N represent the number of available databases. These databases are isolated i.e. that operate independently of each other. Further, these N databases can be either homogenous (same hardware configuration) or heterogeneous (different hardware configuration).

2. Disk Capacity (D): Let real vector D represent the disk space capacity for each of the N databases.

3. Compute Capacity (P): Let real vector P represent the compute capacity for each of the N databases. As compared to measuring disk capacity,estimating compute capacity is often a challenge and depends on multiple factors including hardware and software configuration. For the purpose of this paper, we measure compute capacity in terms of maximum query load that a database can handle. In the case of heterogeneous database system, an optimal query load for a database can be estimated using the equation shown below:

$$P_I = \frac{Concurrency_i \; X \; [2]\text{Error! Bookmark not defined.}\; 86400}{AvgQueryRuntime_i} \qquad (1)$$

   Above, $Concurrency_i$ represents observed historical query concurrency and $AvgQueryRuntime_i$ represents the average query runtime on database i to run a query on database i. Thus, P_i represents the average number of queries that a database can handle on a daily basis. In the case of homogenous databases, one can estimate P_i by distributing existing query volume equally across all the databases i.e. $P_i = TotalQueriesPerDay/N$.

4. Data Elements (T): A data element is defined abstractly over here and can indicate a table, rows of a table or columns of a table. Depending on the definition of data elements, the proposed algorithm will either generate partitions similar to data mart, horizontal or vertical partitioning, respectively. As a result, often the definition of data element is constraint by engineering feasibility. For instance, if data elements are described as set of rows then migration of a partially replicated databases from one state to another would involve migrating rows of a table. For the explanation purpose, we here defined data elements as individual tables and represent the set of tables as T. Thus T_i represents a specific table i.

5. Data Element Size (O): Measured in the same unit as D, O_i captures the disk space occupied by the data element i.

6. Query Classes (Q): A query involves operating on a set of data elements. Thus, one can group queries based on the set of data elements, i.e. queries using the same set of data elements are grouped together. For instance, if a query q requires three data elements, say

T_1, T_3 *and* T_5, we can represent q by the set of $\{T_1, T_3, T_5\}$and further group all the queries this particular set of data elements into one class, referred in [11] as query class. Let Q represent the set of query classes. Note that query classes can have different number of data elements and can be overlapping.

7. Data Element and Query Class Mapping (C): Let C be a binary matrix that describes mapping between query classes and data elements. Thus, if $C_{\{q,t\}} = 1 \; \forall q \in Q, t \in T$ indicates that query class q requires data element t.

8. Query Class Weight (W): Expressed in the same unit as that of P, W_i represents compute capacity needed for the query class Q_i. Here, P_i is measured in terms of daily query load and, thereby,W_i represents average daily number of queries associated with query class Q_i.

9. Replication factor (r): Canonically the term replication factor is used to indicate the minimum number of replicas of data elements. However, we consider it an aspect of a query class and represents the minimum number of isolated databases that should be able to handle a given query class. We find this interpretation more suitable as it ensures not only that data elements are replicated at-least r times but that each query class is covered by at-least by$r$ databases.

Apart from the above variables, we have two decision variables X and Y. X, a binary matrix, represents assignment of data elements to databases;$X_{t,n} = 1$ indicates that the data element t is assigned to a database n, and $X_{t,n} = 0$ otherwise. Y, also a binary matrix, represents assignment of query classes to databases. Similar to X, $Y_{q,n} = 1$ indicates query class q assigned to database n, and $Y_{q,n} = 0$ otherwise. Additionally, we define X' as the existing state of assignment of T databases to N databases. This will be used to determine the migration cost by comparing X to X'.

3.2. Cost Function

As discussed above, four factors influence the design of partially replicated isolated databases: database availability, compute scalability, storage scalability and migration cost. Database availability is a function of the replication factor. As the replication factor increases, database availability increases and vice-versa. In our approach, we assume that database availability is a given parameter and hence one doesn't need to explicitly model for database availability. For the remaining three factors, we propose a simple mixed integer quadratic constraint formalization. Minimizing this formalization provides an optimal partitioning and allocation strategy to uniformly distribute compute and storage utilization across all the available databases. As explained below, the proposed formalization is a linear combination of three individual cost functions associated with the compute scalability, storage scalability and migration cost.

1. Storage Utilization: One of the primary motivations of a partially replicated isolated databases is to minimizeoverall storage consumption across N databases. In practice, however, another important constraint is that disk utilization across N databases to be uniformly distributed. Disk utilization of a database is described as the percentage of the total disk space that is consumed. Given the data element assignment matrix X, the percentage disk utilized$s_i$on a database i can be computed as:

$$s_i = \frac{\sum_{t=1}^{|T|} X_{t,i} O_t}{D_i} \; \forall \; t \in T \tag{2}$$

Since X is a binary matrix, the numerator in the above equation represents the sum of disk space required by the data elements assigned to database i. Normalized by total disk space capacity of database i, s_i represents the percentage of disk space utilized by the all the data elements assigned to database i.

2. Compute Utilization: Compute scalability is achieved by evenly distributing the query load across N databases in proportion to individual databases compute capacity P. Thus, compute utilization of database is described as the percentage of compute capacity that will be utilized by queries assigned to the given database. Given the query assignment binary matrix Y, the percentage query load l_i on a database $i$compared to its compute capacity $P_i$can be computed as:

$$l_i = \frac{\sum_{q=1}^{|Q|} Y_{q,i} W_q}{P_i} \; \forall \; q \in Q \tag{3}$$

One caveat with the above the cost function is that it assigns the complete weightW_q of query class q to a single database. However, in practice, queries associated with a particular query class can addressed by r databases. However, we prefer the above formalization as it models the worst case scenario where only one of $r$databases that can handle a given query class is available and thereby has to deal with the complete load of query class q.

3. Migration Cost: Migration cost is associated with the cost of moving from one state of partially replicated databases to another, i.e. migration from state X' to X. Migration involves two types of operations on databases: writing new data elements that are in X but not in X' and deleting data elements that were in X' but not in in X. Since writing is a much more expensive operation as compared to deleting, we define migration cost based on the cost of writing. Further, the cost of writing is directly proportional to the size of the data element. Hence, as shown in eqn. 5, we define the migration cost as the total size of new data element that will be written on a database normalized by the acceptable amount of writing $\Delta_i$on database i, as defined by the database admin. Normalizing migration cost by Δ helps in two ways. First, it helps database admin enable control the influence of the migration cost. Second, it makes the cost function unitless and therefore comparable to other cost functions (eqn. 3 and 4).

$$m_i = \frac{\sum_{t=1}^{|T|} O_t \left(1 - X'_{t,i}\right) X_{t,i}}{\Delta_i} \tag{4}$$

In the above expression, $\left(1 - X'_{t,i}\right) X_{t,i}$ identifies new data elements that were previously not present on database i but are required in the new state X.

Based on the above three individual cost function, we now define our objective function as minimizing linear combination of the maximum of each of the individual cost functions, i.e.:

$$J(X,Y) = \min (S + L + M) \quad (5)$$

where

- $S = \max (s_1, s_2, \dots, s_N)$
- $L = \max (l_1, l_2, \dots, l_N)$
- $M = \max(m_1, m_2, \dots, m_N)$

subject to:

1. $X_{t,n} \in \{0,1\} \forall\, t \in T, n \in [N]$
2. $Y_{q,n} \in \{0,1\} \forall\, q \in Q, n \in [N]$
3. $\sum_{n=1}^{N} Y_{q,n} \geq r$
4. $X_{t,n} \geq Y_{q,n} \forall\, q \in Q\,, t \in T, n \in [N]$

In practice, one prefers a state with minimal variance in terms of percentage disk utilization and compute utilization relative to individual databases' disk and compute capacity. However, minimizing variance doesn't guarantee optimal utilization of resources. For instance, assuming N homogeneous databases, zero variance can be achieved by replicating all the data elements across all the databases. However, such a state will be far from an optimal solution. Hence, along with minimizing variance, one also need to minimize total disk utilization. Instead, as defined by S, minimizing maximum disk utilization across N databases achieves both the objectives. The same argument goes for L and M.

Another important aspect of our objective function shown in eqn. 6 are the four constraints. While constraint 1 and 2 ensure that the data element X and query class Y assignment matrix are binary, constraint 3 ensures database availability by making sure that each query class is assigned to at-least r databases. The fourth constraint ensures that a query assigned to a database i can only be successfully served if all the data elements required the query are available on database n. Also, note that constraint 3 and 4 together ensure that each data element is replicated across r databases. If desired, similar to constraint 3, one can also add a constraint to enforce a different replication factor, say r', for data elements.

We emphasize that the binary matrix$Y$ reproduces the set of databases that could process a query class, rather than the actual assignment of query workload across databases. This assignment is handled dynamically to load balance with knowledge of the current system load, rather than statically.

4. RESULTS

With millions of Uber trips every day, Uber infrastructure handles petabytes of data. As shown inFigure 2, the data is streamed through Apache Kafka and after that stored in the Hadoop Distributed File System (HDFS). Next, Hive is used to clean and create a modeled data. Data analysis on HDFS using Hive or Presto is usually slow, and hence core business data is further copied to a system of fully replicated isolated Vertica databases. Vertica is a proprietary in-

memory database that provides near real-time interactive query experience; 90% of queries complete within 20 seconds as compared to about a minute in the case of Presto.

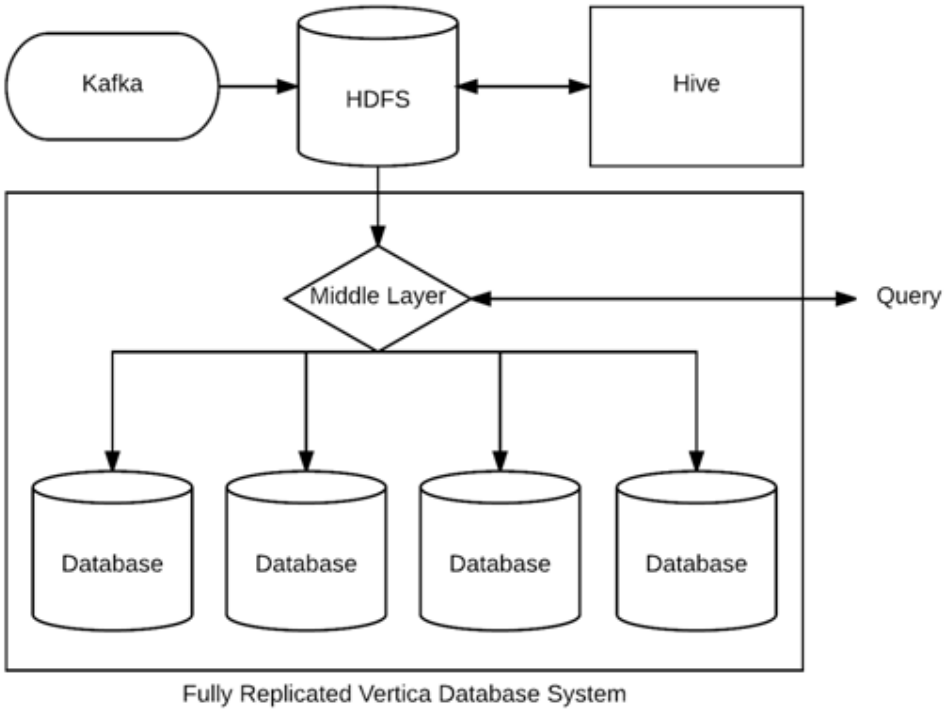


Figure 2. Existing data infrastructure setup at Uber. Data flows from Kafka to HDFS and eventually replicated across multiple isolated Vertica databases. Clients connect to the databases through a middle layer that helps distribute query load across available databases.

In the case of Uber, the compute scalability requirement mainly drives the need for multiple fully replicated isolated Vertica databases system. But, as discussed before, this compute scalability comes at the cost of storage scalability. Apart from optimizing for database availability and compute scalability, achieving a high degree of storage scalability is critical for Uber for two reasons. First, as discussed in Section 1, the additional storage required due to full replication of all the data elements cost millions of additional dollars in terms of licensing fee. Second, there is a limit on how much data can be managed by a single Vertica database to provide near real-time interactive query experience. To address these two major issues, we explored the option of partially replicated isolated databases. The architecture diagram below shows the major component of our solution.

As compared to Figure 2, one key difference is an addition of “data element assignment manager” that informs other components about the placement of the data elements to different databases. The middle layer communicates with the data element assignment manager to determine and route the query to one of the candidate databases that contains all the data elements needed by the query. Every week, based on last four weeks of historical queries, the data element assignment manager generates a new assignment matrix. The assignment matrix is then communicated to the data loader which is responsible for migrating databases from the existing state to the new state.

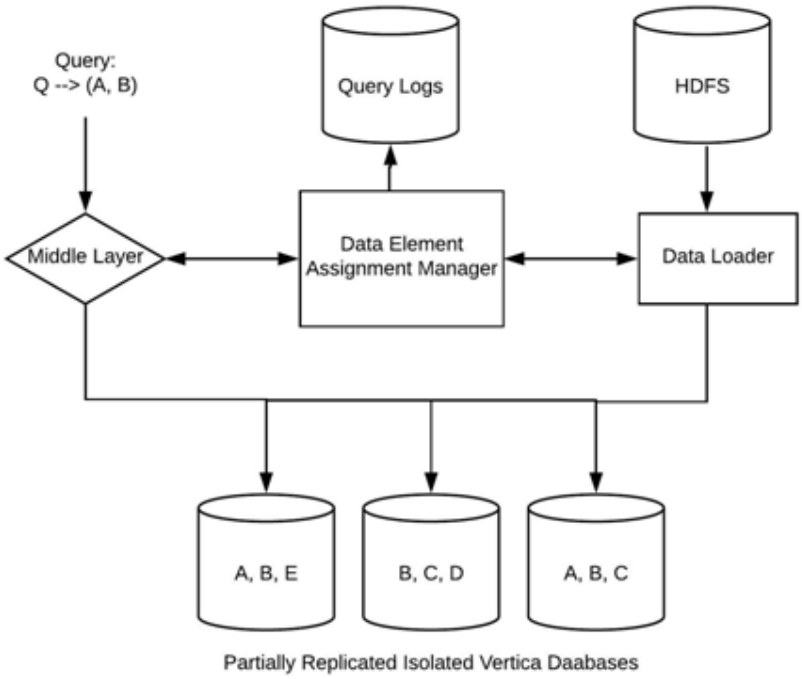


Figure 3. Architecture of our ephemeral partially replicated isolated Vertica databases. The key component is data element assignment manager that re-computes optimal assignment for each data element every week.

In order to evaluate our approach, we ran a ten-week simulation on historical queries. For the simulation, we assumed five homogenous Vertica databases (i.e. N = 5) and a replication factor of 2 (i.e. r = 2). As described in Section 3.1, historical queries are grouped into different query classes based on the set of referred tables. For each query class, we further used the average daily number of queries based on the last four weeks of historical queries as a representation for the query class weight. Further, we approximate compute capacity of each database P_i to be equal to the total query workload distributed equally across N databases i.e. $P_i = \frac{\sum_{i=1}^{|Q|} W_i}{N}$

One of the challenges we faced was the scale of the optimization problem formulated in eqn. 6. With hundreds of tables and thousands of query classes, solving a mixed integer quadratic constraint optimization problem with thousands of decision variables is a non-trivial task. To scale down the problem, we relied on empirical observations to make the optimization problem formulated in eqn. 6 substantially simpler, thereby obtaining a near optimal solution within a few minutes using the commercial mixed integer linear programming solver, Gurobi[8]. First as shown in figure 4, the top 10% biggest tables account for almost 90% of the total data. Thus most of the disk efficiency comes by an optimal placement of these big tables. Based on this observation we define our set of data elements T to include only these top tables and thereby significantly reduce the size of the X matrix. The remaining 90% of the tables are fully replicated as the potential gains in terms of disk utilization are insignificant.

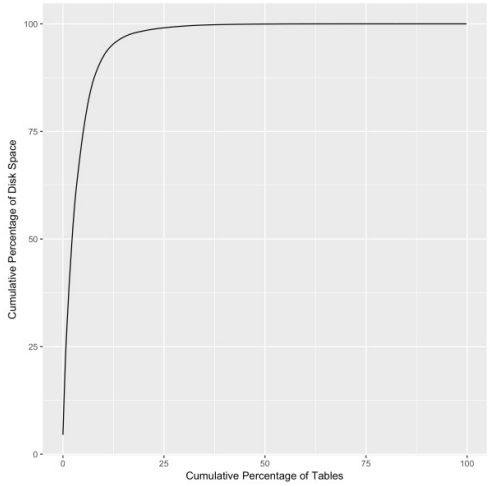


Figure 4. Cumulative percentage of data by percentage of tables in descending order by size. Thus, biggest 10% account for almost 90% of the total data.

Second, a side effect of restricting the set of data elements to the top 10% biggest tables is that it significantly reduces the number of query classes one has to consider. Since query class definition is based on data elements in T, any data element not in T is dropped from the query class definition. If all the data elements needed by the query class are outside the scope of T, then we drop the query class itself from consideration. This is justified since data elements outside of T are fully replicated across all the databases and hence the queries can be executed on any of the N databases. Figure 5 shows the number of query classes as a function of the percentage of tables included in T. Thus, by considering only the top 10% biggest tables (the same as the set tables considered within our first improvement), the number of query classes decreases from almost 100K to 40K. This significantly reduces the size of the Y matrix.

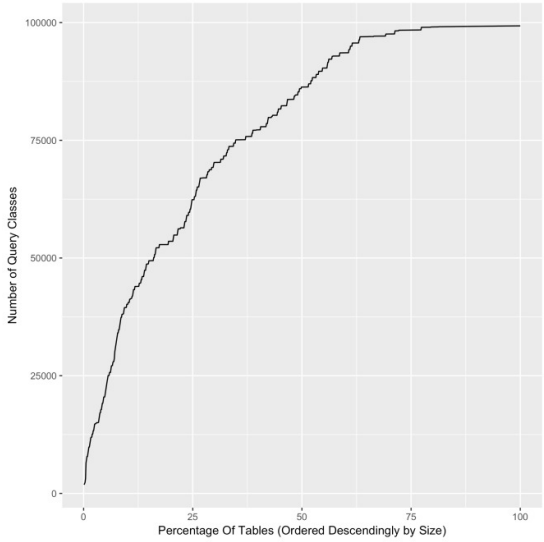


Figure 5. Number of query classes as a function of number of tables considered as part of T. Thus, if we consider the top 10% biggest tables as part of T, the total number of query classes drop from almost 100K to 40K.

Apart from the above considerations, there are new tables that require special handling. Due to a lack of historical data, it is difficult to assign new tables to databases in an optimal way. Hence, for the first two weeks since the creation of a table, we fully replicate new tables across all the databases. This approach helps reduce migration cost associated with changing query patterns related to new tables. From the third week, new tables are treated as regular tables and considered for optimal allocation only if it makes it into top 10% biggest tables.

For the simulation, we generated a new data element assignment matrix X at the beginning of each week and evaluated the performance of our recommended assignment based on proceeding weeks queries. As described below, we measured the performance of the assignment on four different criteria:

1. Savings In Disk Utilization

One of the main objectives of a partially replicated isolated database is to minimize percentage disk utilization of individual databases. Since, our objective is to measure savings in disk space as compared to a fully replicated databases, we define savings in disk utilization as:

$$DiskUtilization_i = 1 - \frac{\sum_{t=1}^{|T|} X_{t,i} O_t}{\sum_{t=1}^{|T|} O_t} \quad (6)$$

In eqn. 7, the numerator represents the size of data elements assigned to database i. The denominator represents the total size of all the data elements. Based on the simulation results, we observed that the median percentage disk savings on an individual database ranged between 58% and 62%. Thus, as compared to fully replicated database, we are able to recover almost 40% disk space. Additionally one can notice from the range that the disk savings is almost evenly distributed across databases.

2. Migration Cost:

As described in Section 3.2, migration cost is concerned with the size of new data elements that were previously not assigned to a given database and are required in the new state and can be computed as:

$$MigrationCost_i = \sum_{t=1}^{|T|} (1 - X'_{t,i}) X_{t,i} O_t \quad (7)$$

Over the 10 week simulation, the median migration cost for the five databases ranged from 0.07 TB to 1.5 TB. This was well within our acceptable level of migration Δ that was set to 5TB. Note that the migration cost for the first week is zero because we are migrating from the fully replicated database to partially replicated database and hence one has to only delete data elements in order to move to a new state.

3. Query Load:

It refers to the percentage of queries from the following week that will be assigned to a database. While, in the actual implementation, the placement of a query is influenced by the current load on candidate databases, for the simulation we randomly assigned the query to one of the candidate databases. The median query load for the five databases ranged from 19.14% to 21%.

4. Dropped Queries:

One of the dangers of a partially replicated isolated database is the lack of all the required data elements by a query on a single database. In this case, the query cannot be satisfied by any of the databases. Over the 10 week simulation, the maximum number of queries dropped in a week was 11. As compared to almost half a million queries that were successfully handled, missing 11 queries was insignificant and not a concern.

5. CONCLUSION

Traditionally, the design of partially replicated databases has focused on increasing storage scalability but without accounting for the migration cost. In this paper, we formalized the notion of the migration cost and presented a new cost-based objective function that along with other factor optimizes the allocation of data elements for migration as well. Based on 10-week simulation results on actual query load on the Vertica database, we demonstrate that this migration based formalization not only helps achieve significant savings in terms of disk utilization but also optimizes for migration. As compared to a fully replicated isolated databases, disk utilization dropped by almost 40%. Further, the median migration cost ranges from 0.07 TB to 1TB across different databases. Although we did observe 11 dropped queries in a week, the number of dropped queries were insignificant as compared to almost half a million queries that our system is able to handle correctly and all the other potential gains in terms of disk utilization.

REFERENCES

[1] Agrawal S, Narasayya V, Yang B (2004) Integrating vertical and horizontal partitioning into automated physical database design. In: Proceedings of the 2004 ACM SIGMOD international conference on Management of data. ACM, pp 359–370

[2] Bernstein PA, Goodman N (1983) The failure and recovery problem for replicated databases. In: Proceedings of the second annual ACM symposium on Principles of distributed computing. ACM, pp 114–122

[3] Bonifati A, Cattaneo F, Ceri S, Fuggetta A, Paraboschi S (2001) Designing data marts for data warehouses. ACM transactions on software engineering and methodology 10:452–483

[4] Ceri S, Negri M, Pelagatti G (1982) Horizontal data partitioning in database design. In: Proceedings of the 1982 ACM SIGMOD international conference on Management of data. ACM, pp 128–136

[5] Coulon C, Pacitti E, Valduriez P (2005) Consistency management for partial replication in a high performance database cluster. In: Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on. IEEE, pp 809–815

[6] Curino C, Jones E, Zhang Y, Madden S (2010) Schism: a workload-driven approach to database replication and partitioning. Proceedings of the VLDB Endowment 3:48–57

[7] El Abbadi A, Toueg S (1985) Availability in partitioned replicated databases. In: Proceedings of the fifth ACM SIGACT-SIGMOD symposium on Principles of database systems. ACM, pp 240–251

[8] Gurob Inc (2016) Gurobi Optimizer Reference Manual. In: Gurobi. http://www.gurobi.com

[9] Kemme B, Alonso G (2000) Don't Be Lazy, Be Consistent: Postgres-R, A New Way to Implement Database Replication. In: Proceedings of the 26th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., pp 134–143

[10] Kemme B, Jiménez-Peris R, Patiño-Martínez M (2010) Database replication. Synthesis Lectures on Data Management 5:1–153

[11] Rabl T, Jacobsen H-A (2017) Query centric partitioning and allocation for partially replicated database systems. In: Proceedings of the 2017 ACM International Conference on Management of Data. ACM, pp 315–330

[12] Schiper N, Schmidt R, Pedone F (2006) Optimistic algorithms for partial database replication. In: International Conference On Principles Of Distributed Systems. Springer, pp 81–93

[13] Song S (2017) A Framework for Design of Partially Replicated Distributed Database Systems with Migration Based Genetic Algorithms. International Journal of Database Management Systems 9:01–21

AUTHORS

Ritesh Agrawal is a lead data scientist at Uber with more than 10 years of experience in the field of big data and machine learning. At Uber, he focuses on using machine learning and artificial intelligence techniques to help scale Uber's infrastructure. Before Uber, Ritesh worked on predictive and ranking models at Netflix, AT&T Labs and Yellow Pages, Inc. He received MS in the field of Transportation Engineering from the University of Illinois at Urbana Champaign (UIUC) in 2004 and thereafter worked towards his PhD in the field of Environmental Earth Sciences from Pennsylvania State University (PSU). His PhD thesis focused on computational tools and technologies such as ontologies and concept maps.

Peter Frazier is an Associate Professor in the School of Operations Research and Information Engineering at Cornell University. He works on problems in learning and decision-making in which decisions affect the data available. This includes Bayesian optimization, multi-armed bandits, Bayesian sequential experimental design, optimization via simulation, and incentivized exploration. He is also a Staff Data Scientist at Uber, where he works on pricing and marketplace design. He received a Ph.D. in Operations Research and Financial Engineering from Princeton University in 2009. He is the recipient of an AFOSR Young Investigator Award and an NSF CAREER Award, and is an associate editor for Operations Research, ACM TOMACS, and IISE Transactions.

INTENTIONAL BLANK

COMPARING RESULTS OF SENTIMENT ANALYSIS USING NAIVE BAYES AND SUPPORT VECTOR MACHINE IN DISTRIBUTED APACHE SPARK ENVIRONMENT

Tomasz Szandala

Department of Computer Engineering
Wrocław University of Technology, Wrocław, Poland

ABSTRACT

Short messages like those on Twitter or Facebook has become a very popular opinions sharing tool among Internet users. Therefore micro blogging web-sites are nowadays rich sources of data for opinion mining and sentiment analysis. However it is challenging because of the limited contextual information that they normally contains. Furthermore the greatest benefit can be achieved by collecting sentiment class in real time - when the post is published, in order to react as soon as possible. Nevertheless, most existing solutions are limited in centralized environments only. thus, they can only process at most a few thousand tweets. Such a sample, is not representative to define the sentiment polarity towards a topic due to the massive number of tweets published daily. Sample analysis has been performed using Machine Learning methodologies alongside with Natural Language Processing techniques and utilizes Apache Spark's Machine learning library, MLlib, on a labelled (positive/negative) corpus containing 4234 tweets regarding Presidential Election in USA in 2016. The analysis has been completed using distributed Apache Spark environment with simulated stream of data from Kafka database.

KEYWORDS

Apache Spark, natural language processing, sentiment analysis

1. INTRODUCTION

Sentiment analysis is a group of methods, techniques, and tools about detecting and extracting subjective information, such as opinion and attitudes, from any text or voice source. [1] Traditionally, sentiment analysis has been about opinion polarity, i.e. whether someone has positive, neutral, or negative attitude towards something or someone. The best applications for attitude recognizing include market survey about new products or political parties/candidates pre-voting support.

Any research appears more attractive when its results can be compared to real life results in whole population. This is why this paper is being focused on Presidential Election in USA in 2016, because scientifically obtained results can be compared to actual results that gave White House to Donald Trump.

David C. Wyld et al. (Eds) : AI, CSTY, NMOCT - 2018
pp. 15–21, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81402

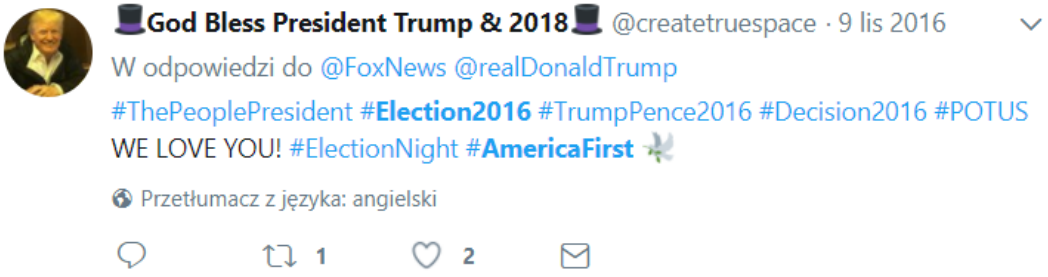
Twitter company has to processes on average 8000 tweets per second (and over 12000 in peak hours) [2] before publishing them for public access. They analyze all data with this extreme fast rate, to ensure that every tweet is following agreement policy and restricted words are filtered out from the messages. All this analyzing process has to be done in real time to avoid delays in publishing tweets live. To analyze such huge data it is required to use some kind of analysis tool. This paper chooses an open source tool Apache Spark. Spark is a cluster computing system from Apache Foundation focused on analyzing any data in parallel[3].

Spark MLlib (Machine Learning Library) is a distributed machine learning framework on top of Spark's Core that, due in large part to the distributed memory based Spark architecture, is as much as nine times as fast as the disk-based implementation used by Apache Mahout (according to benchmarks done by the MLlib developers against the Alternating Least Squares (ALS) implementations, and before Mahout itself gained a Spark interface), and scales better than Vowpal Wabbit[4]. Many common machine learning and statistical algorithms have been implemented and are shipped with MLlib which simplifies large scale machine learning activities.

And last, but not least comes the Apache Kafka database. In this paper it is used to simulate stream of tweets. Even tho Twitter is sharing quite versatile API for fetching specific tweets, the environment in which research has been performer, suffered from unstable network connection which forced emergency measures.

2. TWEET ANALYSIS

A typical tweet data consists of up to 140 characters (as of 7 November 2017 the amount has been doubled). Apart from main text the tweet usually have emoticons and hashtags.



The tweets were chosen by fetching all tweets with hashtags "#Election2016", "#TrumpPence2016" "#Clinton2016", "#POTUS" and few more recommended by Twitter itself. Of course all analyzed tweets came from between 1 June 2016 and 10 November 2016.

Tweets has been at first divided into two groups: Trump or Clinton. Assignment has been made by recognizing any reference to specific candidate like name, surname, vice president surname or motto, like "#AmericaFirst" which can be indisputably connected to one of them. If the tweet contained references to both groups or to none of them it has been rejected from further processing In the next step all messages has been preprocessed. I have converted all of our text to lower-case, removed hashtags from the start and end of the message. I have taken an assumption that surrounding hashtags does not carry any sentiment value, while inner ones (e.g. "#proud") can be valuable in analysis. Since Twitter is very casual, people often include multiples of the same letters in a word, such as “Weeeee wooooooon.” To handle these cases, repeating characters has been reduced to no more than two of the same consecutive letters. All URLs and user names were also discarded since URLs and user names are rather not related to the emotion expressed in the tweet. Furthermore, a standard list of common words that do not express sentiment has been

skipped. The list includes words like “be,” “at,” “the,” etc, which might only slow down analysis performance.

Emoticons were also taken into consideration and were translated into corresponding words (e.g. :-) into "happy"). For the final model for analysis I have chosen simple bag-of-words model [5,6]. It is a simplifying representation used in natural language processing and information retrieval. Also known as the vector space model. In this model, a text (such as a tweet) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. All other models, even tf-idf, which is considered to be more accurate, has no point in tweets analyzing since messages are too short to see full benefit from them.

3. TOOLS FOR DISTRIBUTED DATA ANALYSIS

3.1. General Format, Page Layout and Margins

As I have mentioned in the introduction Twitter has to solve few thousands messages per second. Since adding any advanced analysis to text, like sentiment, may introduce delays in receiving verdict there is a need for an efficient tool. This problem was noticed by Matei Zaharia who proposed in his PhD [7] a distributed platform for parallel data analysis. Apache Spark is a distributed computing framework which performs operations on the working data sets, which are called resilient distributed datasets (RDDs)[3].

Apache Spark’s programming model is based on processing those RDD objects in a form of acyclic data flow, using a set of operators called transformations. Such model supports greatly operations derived from functional programming like mapping, filtering or reducing. Functional paradigms allows to easily split the computation that would be normally performed in a single thread way over multiple cluster nodes. To get the most benefits from this parallelism, the data should be first placed in distributed file system like Hadoop Distributed File System. Even though Apache Spark is written in Scala (functional version of Java) and is run on Java Virtual Machine, it also natively supports Python scripting for data analysis.

3.2. Apache Kafka database

Apache Kafka database is an open-source stream processing platform developed by the Apache Software Foundation written in Scala. The main features of the project is to provide a unified, high-throughput, low-latency platform for handling real-time data feeds. Its storage layer is essentially a "massively scalable pub/sub message queue architected as a distributed transaction log,"[8] making it extremely valuable for enterprise infrastructures to process streaming and/or distributed data.

4. CLASSIFICATION METHODS

4.1. Naive Bayes classifier

One of simplest classification methods is naive Bayes classifier[9,10]. Naive Bayes is a simple multiclass classification algorithm with the assumption of independence between every pair of features. Naive Bayes can be trained very efficiently even on small size of training data. Within a single pass to the training data, it computes the conditional probability distribution of each feature given label, and then it applies Bayes’ theorem to compute the conditional probability distribution of label given an observation and use it for final prediction in validating set.

4.2. Linear Support Vector Machine

Since we are considering only two classes of possible verdict - positive or negative, we can choose binary classifier for sentiment analysis. Support Vector Machines (SVMs, also support vector networks) is supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis [10, 11]. Given a set of training examples, each marked as belonging to one or the other of two categories.

$$(x_1 ,y_1),(x_2 ,y_2),. . . (x_n ,y_n) \quad (1)$$

Where (eq. 1):

x_i are the vectors of features and y_1 is on of binary class label: -1 or +1. We want to find the "maximum-margin hyperplane" that divides the group of points for which y=-1 from the group of points for which y=1, which is defined so that the distance between the hyperplane and the nearest point from either group is maximized. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

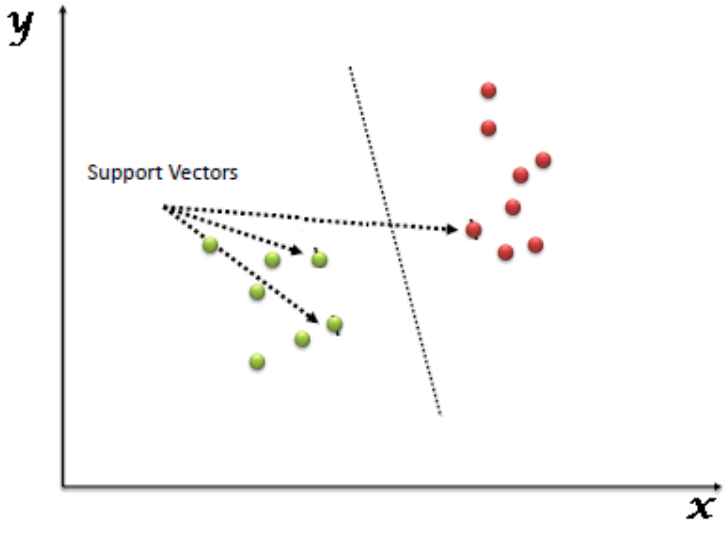


Fig. 1

Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line)

5. THE EXPERIMENT

5.1. Training

The source tweets were shuffled and splitted up into training and test sets in-order to validate the trained model. The division for this work is carried out in the ratio of 3:1. This means: 3176 of the messages are used for training and rest of the data is left for later testing. It is also worth to mention that around 70% of tweets were negative. At this point the target of the tweet (Trump or Clinton) were not taken into consideration.

5.2. Validating

The measure of classifier performance was accuracy, the amount of correctly recognized messages divided by quantity of all messages taken into consideration. The process of training and validating has been repeated five times and the most promising pair of classifiers (the one with highest accuracy) will be taken into classifying real data.

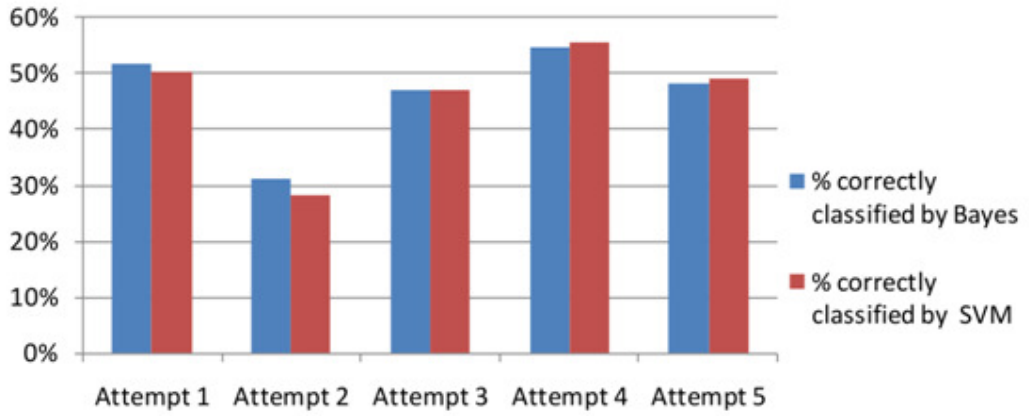


Fig. 2 Comparison of correctly classified tweets between attempts

The results (fig. 2.) shows that attempt fourth is the most accurate one for both classifiers. Bayes classifier was 54,7% time right and SVM achieved 55,6% accuracy. However the second attempt shows distinctive results. After closer look I have noticed that after shuffling almost all positively labeled messages fell into validating set and since training set was extremely unbalanced we can see biased results.

5.3. The real data

Last step of the experiment was to use the best fitted classifiers for analyzing sentiment of the real data. At first the 40 000 tweets with hash tag, associated with elections, were downloaded and stored into Kafka database. 20 000 regarding Trump, 20 000 about Clinton. Afterwards the analysis has been performed using previously trained classifiers and Apache Spark streaming on single node. In the next iterations I have added one and later second slave node to Apache Spark.

Table 1 Sentiment analysis on randomly chosen 20 000 tweets per candidate

	Trump positive	Trump negative	Clinton positive	Clinton negative
Bayes	9 595	10 405	7 378	12 622
SVM	8 998	11 002	6 841	13 159

Since real tweets have no label about its sentiment analysis we cannot judge classification accuracy. Nevertheless dominating negative sentiment (table 1.) may be prove for common belief that American citizens were voting for candidate they disliked less[12].

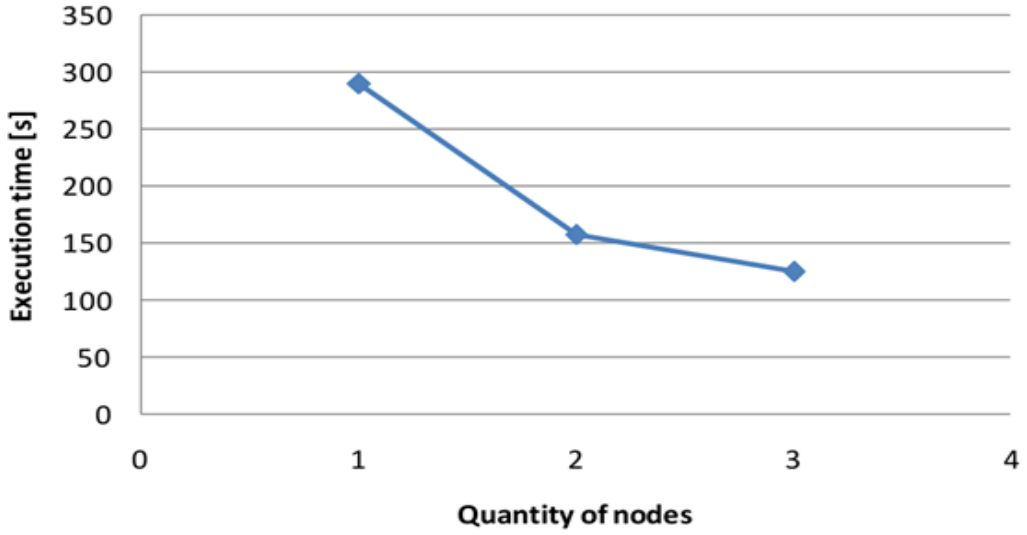


Fig. 3 Comparison of time of execution on 1, 2, 3 single thread Apache Spark nodes

Introducing parallel processing provides meaningful speed up in data processing as we can see in fig. 3. There is around 45% improvement on execution time if we add second node and close to 60% in total if we split our computation on 3 executors.

6. CONCLUSIONS

Papers in this format must not exceed twenty (20) pages in length. Papers should be submitted to the secretary AIRCC. Papers for initial consideration may be submitted in either .doc or .pdf format. Final, camera-ready versions should take into account referees' suggested amendments.

6.1. Naive Bayes vs SVM

While comparing accuracy of Naive Bayes classification and Support Vector Machine we do not see significant difference between them. Only the most important drawback of SVM is that we can compare only two classes with one such classifier. If we choose to set sentiment as positive or negative there will be no issue in it. But if we add for example third class "neutral" we would have to consider 3 instances of SVM classifier: each for each pair of classes and then use voting mechanism to choose the best fitting one.

6.2. Distributed processing in Apache Spark.

Introducing parallel processing feature from Apache Spark we can notice significant improvement in time of execution as we add more nodes. Furthermore Apache Spark ensures continuity of work even if one of the nodes disconnects. It rebalances to load to other working node(s) as long as at least one is up. At worst we can lose only data that was being processed at the moment of downfall.

6.3. Application in real data

As mentioned above: we can conclude that negative opinions dominated in 2016 Elections. Unfortunately twitter analysis show clearly that Trump should have easily won the elections. While he indeed won the election it happened only due to American specific electoral voting

system. In raw numbers Donald Trump would have lost to Hillary Clinton 46% to 48% votes. This means that twitter population is not fully representative and even if sentiment analysis would be perfected we cannot rely only on it.

REFERENCES

[1] M. V. Mantyla, D. Graziotin, M. Kuutila: The evolution of sentiment analysis A review of research topics, venues, and top cited papers, Computer Science Review, Volume 27, 2018.

[2] [online] Available: http://www.internetlivestats.com/twitter-statistics.

[3] Zaharia, M., Chowdhury, M., et al.: Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. USENIX Symp. Networked Systems Design and Implementation

[4] Sparks, E.; Talwalkar: Spark Meetup: MLbase, Distributed Machine Learning with Spark. slideshare.net. Spark User Meetup, San Francisco, California , 2014.

[5] Youngjoong K.:"A study of term weighting schemes using class information for text classification SIGIR'12. ACM, 2012.

[6] Mikolov T., Chen K., Corrado G., Dean J.: Efficient Estimation of Word Representations in Vector Space 2013.

[7] Zaharia, M.: An Architecture for Fast and General Data Processing on Large Clusters. University of California, Berkeley. 2015.

[8] Mouzakitis E.: Monitoring Kafka performance metrics, www.datadoghq.com/blog, 2016.

[9] Russell S., Norvig, P.: Artificial Intelligence: A Modern Approach (2nd ed.). 2003

[10] Rish, I.: An empirical study of the na ive Bayes classifier. IJCAI Workshop on Empirical Methods in AI, 2001.

[11] Cortes C., Vapnik V.: Support-vector networks. Machine Learning. 20 (3), 1995.

[12] Long H.: Voters say this is the ultimate 'lesser of two evils' election, 2016.

AUTHORS

Tomasz Szandala – PhD candidate From Wroclaw, Poland, involved in multiple projects in the field A.I. Apart from university a DevOps engineer at NOKIA.

INTENTIONAL BLANK

DYNAMIC INFERENCE OF PERSONAL PREFERENCE FOR NEXT-TO-PURCHASE ITEMS BY USING ONLINE SHOPPING DATA

Yun-Rui Li[1], Ting-Kai Hwang[2] and Shi-Chung Chang[1]

[1]Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan
[2]Department of Journalism, Ming Chuan University, Taipei, Taiwan

ABSTRACT

With more and more people shopping online, companies deal with customer data input not only in high volume but also dynamic. In order to attract target customers more effectively and to provide customers with more personalized services, how to automatically extract personal preference from the real-time data and make real-time recommendation has been growing in importance for businesses in the competitive modern society. Current data analysis methods for online shopping recommendation largely rely on historical transaction record. Analyses have indicating that next items a customer would like to buy not only depend on one's past historical records but on the item currently being put into the shopping cart. This paper designs an engine to combine each customer's past transaction and current shopping cart data to dynamically infer one's preference for the next items. The design, Transaction-Data Based Real-time Preference Inference Engine (TRPIE), consists of two innovative ideas. The first exploits the purchasing sequence information and turns one's purchase history into a temporal series of data, where a customer's dynamic purchasing behaviour information lies. The second is a design of a two-layer Recurrent Neural Network (RNN) for extracting personal purchasing preference pattern from the temporal series of data to infer preference of next items. A reference implementation of TRPIE design integrates existing tools such as Keras, tensorflow™, sklearn™, and Mlxtend™. Test results over real data from 1,374 people show that prediction accuracy has doubled that obtained by a basket analysis method, which ignores sequentiality of purchasing items.

KEYWORDS

Online shopping, transaction data, purchasing sequence, dynamic preference inference, recurrent neural network.

1. INTRODUCTION

In order to keep pace with customers, it is imperative for companies to be able to predict customer behavior such that they can cater to the needs of customers in the future [14]. However, customer needs, situations, expectations, and demands constantly change and evolve [21]. So there would be no way to understand them beyond 'today' without some way of predicting customer behaviour. Predicting future customer behavior has thus been an elusive goal. But, today, with the advancements in data analytics, predicting customer behavior has become more achievable than ever before.

David C. Wyld et al. (Eds) : AI, CSTY, NMOCT - 2018
pp. 23–34, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81403

In practice, it's very common to speculate customers' future behavior by predicting individual customer preferences. With customer preferences predicted, retailers may, for instance, entice extra purchases to the target customers. In a study of customer preference [3], the author made a survey on product rankings in order to figure out customer preferences.

In the era of e-commerce, real-time marketing allows brands to engage with their customers based on real-time information, such as their behavior and actions on a website. More specifically, real-time marketing needs to measure customer behavior in real time for adjusting companies' marketing strategies based on their holistic shopping behavior and to offer more relevant services that are individualized and contextual. With digital transformation, automated data collections and increase of computing power, predictive analytics software reliably helps forecast and influence purchasing behavior [18] and more and more companies are adopting it.

E-commerce retailers have a significant advantage over brick-and-mortar businesses when it comes to collecting data about customers' transaction data. Whenever a sale transaction is made, all purchases by a customer are on record through the membership window with information items such as payment type, products purchased, date of purchased, delivery information, etc [17]. Therefore, for online stores, it is rather easy to collect the data. According to [17], customers preferences may be extracted from their high-frequency purchasing behavior. Online groceries are consumable commodities, which will be purchased repetitively and frequently by customers.

In this paper, the problem scenario is on online groceries, where the products belong to daily necessities, are of low-price and frequent and repetitive purchases in specific categories. Online grocery shopping data provides personal behavior information through both historical and current shopping cart data. The requirements for a shopping cart are that the system may collect data real-time from each shopping cart about what products a customer adds into the shopping cart. Such requirements hold in online grocery stores or checkout-free grocery stores, where real-time data is collected via various sensors, computer vision recognition and internet of things technologies..

Figure 1 depicts the specific problem scenario of this paper, where the sequentiality of personal purchasing behaviour and historical and dynamic shopping cart data input are highlighted. Instead of analysing only the statistics from historical data, this paper will, this paper will address how to combine sequential correlation among each customer's implicit and habitual purchasing behind the historical data with current online shopping cart data so as to dynamically infer one's preference for the next-to-purchase items.

This paper will use Instacart Online Grocery Shopping Dataset1 in 2017 as the test data set. Instacart is an American company that operates as a same-day grocery delivery service in the United States [12]. Customers select groceries through a web application from various retailers and delivered by a personal shipper. Instacart makes Instacart Online Grocery Shopping Dataset accessible by the general public, where there are 206,209 customers' transaction history including 3,421,083 orders in total, and it categorizes 49,685 products into 21 departments and 134 aisles.

This paper will design an engine to combine each customer's past transaction and current shopping cart data to dynamically infer one's preference for next-to-purchase items. The design, Transaction-Data Based Real-time Preference Inference Engine (TRPIE), consists of two innovative ideas. The first exploits the purchasing sequence information and turns one's purchase history into a temporal series of data, where a customer's dynamic purchasing behaviour information lies. The idea is motivated by the fact that a customer's purchasing behavior is a dynamic process [16], and the purchasing has correlation in sequence by personal purchasing habit or preference. Current basket analysis [20] does not consider such dynamic aspect. The second is a design of a two-layer Recurrent Neural Network (RNN) for extracting personal

purchasing preference pattern from the temporal series of data to infer preference of next items. A reference implementation of TRPIE design integrates existing tools such as Keras, tensorflowTM, sklearnTM, and MlxtendTM. Test results over real data from 1,374 people show that prediction accuracy has doubled that obtained by a basket analysis method, which ignores sequentiality of purchasing items. more specifically, from an average accuracy of 18.17% to 36.46% by TRPIE.

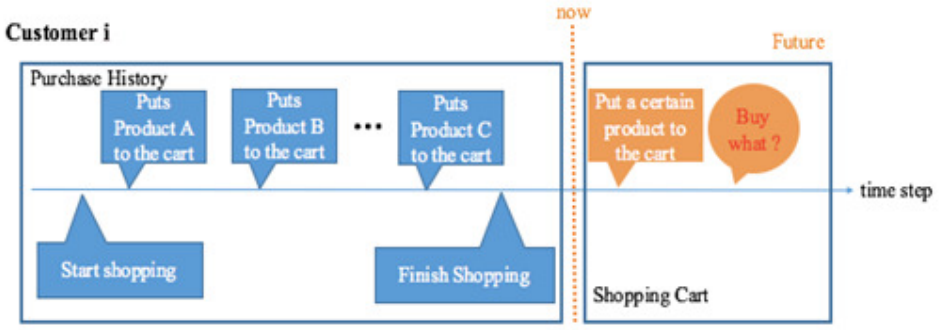


Figure1: Illustration of Problem Scenario

2. DATA ANALYTICS OF CLUSTERING FOR CUSTOMER SEGMENTATION

According to [6], even though we may have data from millions of customers, these customers may only belong to a few segments: customers are similar within each segment but different across segments. For instance, different market segments may have different product preferences and behavioral patterns. Therefore, we may often want to analyze each segment separately, as segments may have different habitual characteristics.

Customer segmentation is the process of dividing customers into groups based on common characteristics so companies can market to each group effectively and appropriately. Segmentation allows marketers to better tailor their marketing efforts to various audience subsets. Those efforts can relate to both communications and product development. Specifically, segmentation helps a company: (1) focusing on the most profitable customers [2], (2) better predict customer's purchasing behavior for upsell and cross-sell3 other products and services [19].

In general, in order to develop refined segment-specific insights, it is often necessary to split our data into segments and perform any subsequent analysis within each segment [6]. Nowadays, clustering techniques are frequently used to do so and do well.

2.1. Data Analytics of Association Rule Learning for Discovering the Connections Between Products

According to [20], association rule learning helps a company: (1) to market precisely (e.g. target customers who buy flour with offers on eggs, to encourage them to spend more on their shopping basket), (2) to drive recommendation engines. Shortly, it not only improves the customer shopping experience but also a company's marketing campaign.

Association rule learning is a rule-based machine learning method for discovering the associations and connections between products in large-scale transaction data [1]. General speaking, market basket analysis is perhaps the most famous example of association rule, and is employed today in many application areas including web usage mining, intrusion detection, and continuous production, etc [1]. In a market basket analysis, marketer always want to see if there

are combinations of products that frequently co-occur in transactions [20]. For example, maybe people who buy flour and sugar, also tend to buy eggs because a high proportion of them are planning on baking a cake. However, unlike sequence mining, association rule learning typically does not consider the order of items within transaction data [1].

2.2. Data Analytics of Sequence Mining for Finding Statistically Relevant Patterns Between Data

Sequential pattern mining is a topic of data mining concerned with finding statistically relevant patterns between data examples where the values are delivered in a sequence. Furthermore, with modern technologies, it enables us to record sequences of online user activity at an unprecedented scale. Although these methodologies mentioned above have been successfully applied in many tasks, it does not take into account the temporal aspect that characterizes sequential data [6]. Specifically, those machine learning models are not suited for sequential data, since they consider each input sample independent from previous ones.

In contrast, given an input sequence, Recurrent Neural Networks (RNNs) are able to process each element at a certain time step, storing the necessary information of each element [15]. RNNs solve various problem with inherently sequential information by connecting the hidden layer with itself [6]. They are able to maintain an internal state which allows them to exhibit dynamic temporal behavior and use the hidden state as an internal memory. Shortly, RNNs process sequences in a natural way.

3. PERSONAL PURCHASING PREFERENCE PATTERN EXTRACTION

In this section, we shall proceed to the Personal Purchasing Preference Pattern (PPPP) extraction method. The goal of this section is to describe how the recurrent neural network (RNNs) is applied to PPPP extraction based on the purchase history data.

3.1. Preparation for Personal Purchasing Preference Pattern Extraction

In this subsection, we will show preliminary work for purchasing preference pattern extraction. Firstly, we show the input of purchasing preference pattern extraction. Secondly, we introduce preparation of training data with target via CPHT algorithm.

3.1.1 Input Data

We extract the following information from the purchase history data, which will be the inputs of purchasing preference pattern extraction:

i1) User ID: It identifies customers, because we want to capture the personal purchasing preference pattern.

i2) Order ID: It consists of Product ID, which tells us which products are bought at this order.

i3) Product ID: To know what products a customer had purchase.

i4) Catalog of Products: It's for the one-hot encoding3 of the specific Product ID.

i5) Add to Cart Order: Order in which each product was added to cart. It's for the RNN to run through the sequence of items consumed by a user, item by item.

Sequential Product Sequence:

Each person's purchase history data can be represented as $\{o^t\}_{t=1}^{n_i-1}$ where o^t is the Order ID at t_{th}. And, the each order o^t within purchase history denotes as product sequences $\{x_j\}_{j=1}^{T}$ where x is product ID and *T* is Add to Cart Order.

Here, we shall give a definition of Personal Purchasing Preference Pattern (PPPP).

Definition of Personal Purchasing Preference Pattern:

According to [9], purchasing patterns reflect how customers purchase goods or service. Like basket analysis, purchasing preference pattern is like a rule that tells us when someone buy product, which product he/she will buy next. In general, personal purchasing pattern like the rule represents a person's purchasing habit or purchasing experience. Also, we focus on customer-oriented strategy and critical marketing, so use individual data to extract pattern. Then, we can predict future behavior via understanding the personal purchasing pattern. Therefore, in the research, we define personal purchasing preference pattern as two-layers RNN model and weights, fi(.).

Definition of Personal Purchasing Preference Pattern Extraction:

How to extract the two-layers RNN model and weights characterizing purchasing preference pattern of each customer based on the above inputs i1) to i5)?

Given a customer i's purchase history data, $\{o^t\}_{t=1}^{n_i-1}$ design function CPHT(.): {all orders within purchase history data}→{input sequences} and {corresponding targets}, called training data with targets, so that RNN(.): {input sequences} and {corresponding targets}→ fi(.) can fit all training data with targets to the model, fi(.). More detailed illustrations such as CPHT(.) and RNN(.) will be addressed in the following.

3.1.2 Data Pre-processing

Now, as we already know, we need the training data with target for follow-up RNNs. So, we need an algorithm to split each order within purchase history data into input sequences and corresponding targets as training data with target, such that the RNNs model can learn to predict products. Because functionality of this algorithm is converting each order within purchase history data into training data with target, we call this algorithm CPHT.

Table 1 below is an example showing the input and output of CPHT(.). CPHT(.) can read multiple orders at the same time and outputs the input sequences and corresponding targets of each order. The interpretation of the input sequences and corresponding targets is one by one explained in Figure 2. Each red circle in Figure 2 is called one example of training data.

Table 1. Input and output from the CPHT(.)

Input:
Each order within purchase history data, o^t. . $o^t:(x_1, \ldots, x_T)$
Output:
{input sequence} and {corresponding targets}, that's usually called training data with targets - {input sequence} : { (x_1) ,(x_1, x_2) . . . , ($x_1, x_2, \ldots, x_{T-1}$) } - {corresponding targets} : { $x_2, \ldots, x_T$}

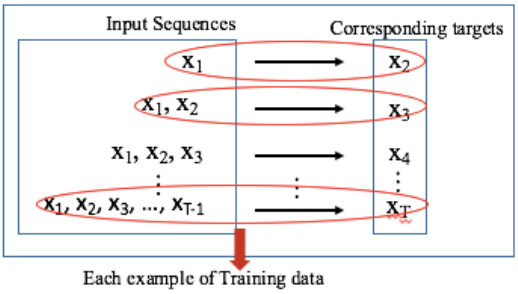


Figure 2. Exemplary of input sequences and corresponding targets from CPHT(.)

Basically, the cores of CPHT are two rules. The first rule is that if there is only one product in this order, the order will not be treated as an example of training data because we cannot know what he will buy after he has bought the product. The second rule is that the last product in each order will not be considered as an example of training. data for the same reason mentioned in rule1.

3.2. Preference Extraction by using RNNs to Exploit Sequential Information Availability

3.2.1 Recurrent Neural Networks

Recurrent neural network is one type of Neural Network architecture: It process sequences by iterating through the sequence elements and maintaining a "state" containing information relative to what they have seen so far [4]. Each person has the individual purchasing behavior process and his historical transaction record can be modeled as sequential data, which has the particular trait that can incorporate a temporal aspect [10]. Then, the process gradually becomes habits or experience [19], representing habitual purchasing behavior. As a result, RNN is naturally applied to extraction of sequential purchasing preference pattern from purchase history data for real-time marketing.

3.2.2 Long short-term Memory (LSTM)

The LSTM model was developed by Hochreiter and Schmidhuber. Primarily, it is for overcoming the problem of vanishing gradients in RNN [11]. The gradient of some of the weights in RNN would become too small or too large if the network is unfolded for too many time steps. This is called the vanishing gradients problem [7]. However, this LSTM model resembles a standard recurrent neural network with a hidden layer, but each ordinary node in the hidden layer is replaced by a memory cell [11] to solve the problem. The number of repeating modules in LSTM is determined by the length of time steps. In short, LSTM is known to learn problems with long-term temporal dependencies, so an LSTM may succeed in the sequence learning.

3.2.3 Supervised Learning and RNNs with LSTM

Supervised learning is by far the dominant form of deep learning today, with a wide range of industry applications, such as sequence generation, syntax tree prediction and object detection, etc. Generally, almost all applications of deep learning that are getting the spotlight these days belong in this scenario.

With supervised learning, humans would give input data and corresponding targets to the computer. After the computation outputs the rules, then these rules could apply new data to produce original targets [4]. Supervised learning mainly could be classed into regression task and classification task depends on the different scenarios. In our research, our task belongs to classification task, and we choose RNNs as our classifier. Because, as mentioned in the early subsection, it is suitable for handling sequential data. For handling the long-term dependencies in the sequences well, our research chooses LSTM model for our RNN to extract purchasing preference pattern from purchase history data.

3.2.4 Purchasing Preference Pattern Extraction

Table 2 below shows input and output of RNN(.) with LSTM; RNN(.) with LSTM can be fed into all examples of training data relevant to our classification task. The following covers what our RNN with LSTM looks like and how to fit all these examples obtained from CPHT(.) to a model via RNN(.) with LSTM.

Table 2. Input and output from RNN(.) with LSTM

Input:
All training data with targets.
Output:
two-layers RNN model and weights, $f_i(.)$, representing customer i's purchasing preference pattern.

We need to understand how the RNN with LSTM processes these examples more clearly, so a step-by-step demonstration is presented through an exemplary training data with targets. RNN with LSTM processes these examples by iterating through the sequence elements and maintaining a "state" containing information relative to what they have seen so far [4].

To conclude, the RNN(.) would come up with model architecture that you assigned and trained weights for automating our classification task after seeing all examples from personal purchase history data. After training, we obtain RNN model with two-layers LSTM and weights for predicting personal preference for next-to-purchase product, fi(.). It represents customer i's purchasing preference pattern in our research.

3.3. Learning Process of RNN

Given the input sequences and corresponding targets, a new method is required to infer preference. The learning process to assess a personal RNN classifier (as shown in Figure 3) from scratch.

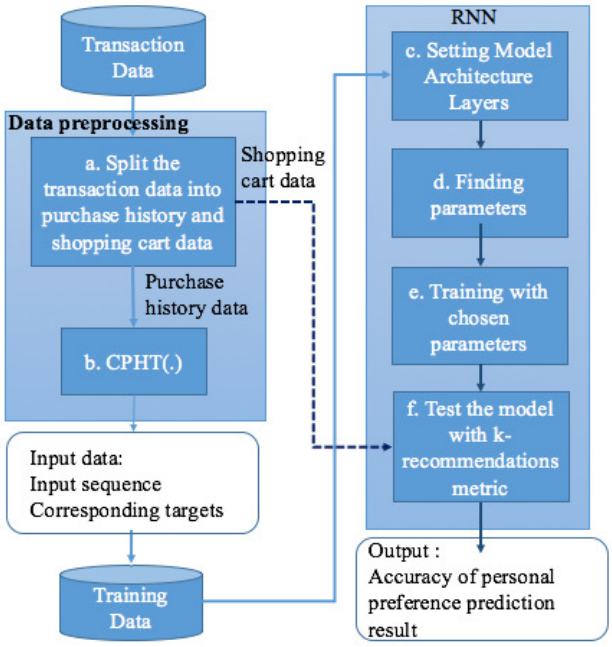


Figure 3. Learning Process of RNN

4. TRANSACTION-DATA BASED REAL-TIME PREFERENCE INFERENCE ENGINE (TRPIE) IMPLEMENTATION AND EXPERIMENT RESULT

This section presents the reference implementation of TRPIE methodology and the performance evaluation results of TRPIE over transaction data that Instacart provides.

4.1. Definition of TRPIE System

To implement TRPIE, there will be a definition of personal recommendation for next-to-purchased product.

Definition of Personal Recommendation for Next-to-Purchase Product:

Given the product that was put into shopping cart by the customer i, $p_{shopping}$, design function TRPIE (.): $p_{shopping} \rightarrow$ {m products that customer i is likely to buy the next}.

4.2. Architecture of Reference Implementation of TRPIE

Based on different methodologies and algorithms presented in previous sections, this section illustrates how these different parts are realized in an integrated system.

The entire system is implemented on Unix-like operating system and programmed in Python 2.7. In this implementation, scikit-learn[1] is adopted to implement clustering techniques. MLxtend[2] is adopted to implement market basket analysis [13]. Keras[3] and Tensorflow[4] is adopted to implement RNN-Based PPPP. All of these existing software are executed in Unix-like operating system. Within the Unix-like system, a Python environment is required to install. Software including scikit-learn[1], MLxtend[2], Keras[3], and Tensorflow[4] are glued together by using python scripts. In Python®, CPHT, GPIE, PPPP extraction, and other Python scripts are built for

processing and analyzing the data are developed. To modularize, a file folder is built. The below 4 shows the mutual relationships between different systems and stacks.

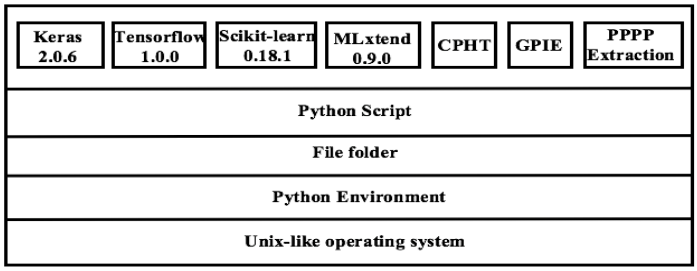


Figure 4. Software stack of TRPIE

4.3. Experimental Validation of TRPIE

The experiment is made based on Instacart Online Grocery Dataset [8]. To obtain statistically meaningful results [4] and more data to enhance power of machine learning-based methods [4], the customers who have purchased less than 99 times during data record period are filtered. At last, leaving 1,374 customers selected and to be analyzed.

Input:

We take data of 100 orders from each customer of these 1,374 customers. We consider the top 97 orders as purchase history data for training and consider the rest 3 orders will be considered as shopping cart data for testing.

Hypothesis:

Because customer's purchasing behavior is a dynamic process, we hypothesized that the RNN-based Personal Purchasing Preference Pattern considering order of purchasing items is able to better reflect dynamic purchasing behavior than something that does not consider. Therefore, we adopt preference for next-to-purchased purchase product inferred from PPPP, given the products that customers put into shopping cart, as prediction on next purchase product.

Baseline - Apriori algorithm [13]:

To the best of our knowledge, there is no comparable personal preference extraction algorithms that exploit purchasing sequence information in the literature. We shall consider the Apriori algorithm as the baseline for performance comparison with PPPP. The reason is that existing Apriori algorithm does not consider the sequential process in customer's transaction data, instead PPPP using RNN considered the sequential process.

Setting for Comparison:

We use the same purchase history data as what PPPP did. That is, the top 97 orders as purchase history data from those 1,374 people, and we uses the top 10 frequent products obtained from individual purchase history data as preference inference through Apriori algorithm (minimum support = 0.1 by default). It's often used in practical applications.

Evaluation Metric:

In this experiment, we use a short-term prediction metric to test the performance of PPPP, which refers to [5]. A short-term prediction aims to predict which product the customer i will purchase the next (i.e. right after the last one). The accuracy of short-term prediction for the performance measurement of PPPP is mathematically defined as follows.

Definition of Accuracy of Short-term Prediction:

$$\text{Individual prediction } accuracy \text{ by } PPPP \equiv \frac{|p_i \cap \{x_i\}|}{|p_i|} \quad (1)$$

Result:

The numerical results are shown in Table 3. PPPP achieves 18.29% higher Individual Prediction Accuracy than Baseline on average and standard deviation is maintained at a similar scale.

Table 3. Statistics of PPPP/ Comparable Baseline Accuracy

Inference Accuracy of 1374 customers	PPPP	Baseline
Mean	36.46%	18.17%
Standard Deviation	20.66%	17.98%
Min	0.00%	0.00%
25%	22.04%	5.00%
50%	31.86%	13.63%
75%	45.45%	25.71%
Max	100.00%	100.00%

5. CONCLUSIONS

This research combines the data of the customer's past transaction record and the items currently being put into the shopping cart to dynamically infer one's preference for the next items. A transaction-data based real-time preference inference engine (TRPIE) was implemented. Also, a two-layer Recurrent Neural Network (RNN) was adopted for extracting personal purchasing preference pattern and inferring preference of the next-item purchase. In addition, the inference ability of TRPIE was demonstrated by experiment over transaction data that Instacart provided.

TRPIE was implemented by integrating different designs of methods and prestigious open-source toolkits such as Keras, tensorflowTM, sklearnTM and MlxtendTM. In the experiment, the results are as expected. PPPP, considering information of shopping cart data and dynamic customer behaviors, achieves 18.29% higher than benchmark on average.

To conclude, this study focused on the research of preference inference. Since the dynamic feature of a customer's purchasing behavior, the item of purchasing has correlation in sequence by personal purchasing habit or preference. Therefore, firms or manufacturers can fine-tune their marketing strategies in real time based on the identified customer preferences.

ACKNOWLEDGEMENTS

This work was supported in part by the Ministry of Science and Technology, Taiwan, ROC, under Grants MOST-106-2221-E-002-129 and MOST-107-2221-E-002-184.

REFERENCES

[1] Agrawal, Rakesh. Imieliński, Tomasz & Swami, Arun (1993) "Mining association rules between sets of items in large databases", Proceedings of the 1993 ACM SIGMOD international conference on Management of data.

[2] Bowes, Pitney. "Customer Segmentation and Profitability", [Online] Available at http://www.pbinsight.com/files/resource-library/resource-files/ppb-cust-segment-and-profit.pdf

[3] Cao, Yu "Marketing research report of Burberry cosmetics", [Online] Available at http://www.academia.edu/6610615/Marketing_Research_Report_of_Burberry_Cosmetics/

[4] Chollet, Francois (2017) Deep Learning with Python (1th Edition), Manning Publications.

[5] Devooght, Robin & Bersini, Hugus (2017) "Collaborative filtering with Recurrent Neural Network," arXiv:1608.07400v2 [cs.IR].

[6] Evgeniou, Theos. "Introduction to data analytics for business", [Online] Available at http://github.com/InseadDataAnalytics/INSEADAnalytics/tree/master/CourseSessions/Sessions45/

[7] Gamboa, John Cristian Borges (2017) "Deep Learning for Time-Series Analysis", arXiv:1701.01887v1 [cs.LG].

[8] Instacart (2017) "Three Million orders, open sourced", [Online] Available at https://www.instacart.com/datasets/grocery-shopping-2017/

[9] Kahn, Barbara (2012) "Buying Pattern", [Online] Available at http://kwhs.wharton.upenn.edu/term/buying-patterns/

[10] Ko, Young-Jun. Maystre, Lucas & Grossglauser, Matthias (2016) "Collaborative Recurrent Neural Networks for Dynamic Recommendation System", JMLR: Workshop and Conference Proceedings 63:366–381.

[11] Lipton, Zachary C. Berkowitz, John & Elkan, Charles (2015) "A critical review of Recurrent Neural Network for sequence learning", arXiv:1506.00019v4 [cs.LG].

[12] Manjoo, Farhad (2015) "Instacart's Bet on Online Grocery Shopping", The New York Times. [Online] Available at https://www.nytimes.com/2015/04/30/technology/personaltech/instacarts-bet-on-online-grocery-shopping.html

[13] Mlxtend (machine learning extensions), [Online] Available at http://rasbt.github.io/mlxtend/

[14] Newman, Emily (2016) "Importance of predicting customer behaviour", [Online] Available at http://corp.yonyx.com/customer-service/17790/

[15] Santolaya, Daniel Sánchez (2017) "Using recurrent neural network to predict customer behavior from interaction data", Master Thesis, University of Amsterdam.

[16] Soloman, Michael R. (2016) Consumer Behavior: Buying, Having, and Being (12th Edition), Pearson Education Limited.

[17] Teng, Hui-Ping (2017) "Infer Individual Customer Preference for a New Product Based on Supermarket Transaction History," Master Thesis, IIE, NTU.

[18] Waxer, Cindy (2010) "What will your customers buy next", MIT Technology Review, [Online] Available at https://www.technologyreview.com/s/421928/what-will-your-customers-buy-next/

[19] Wong, Danny (2017) "How Ecommerce Companies Can Cross-Sell and Upsell to Increase Customer Lifetime Value", [Online] Available at https://conversio.com/deep-dive/cross-sell-upsell-increase-customer-lifetime-value/

[20] Yali (2017) "Market basket analysis identifying products and content", [Online] Available at https://discourse.snowplowanalytics.com/t/market-basket-analysis-identifying-products-and-content-that-go-well-together/1132/

[21] Yeong, BinCho. Yoon, HoCho & Soung HieKim (2005) "Mining changes in customer buying behavior for collaborative recommendations", International Journal of Expert Systems with Applications, Volume 28 Issue 2, pp. 359-369.

COUNTERING TERRORISM ON SOCIAL MEDIA USING BIG DATA

Ali Alzahrani[1], Khalid Bashir Bajwa[2], Turki Alghamdi[3] and Hanaa Aldahawi[4]

[1,2,3]Department of Computer Science,
Islamic University of Madinah, Madinah, Saudi Arabia
[4]Department of Information Science,
King Abdulaziz University, Jeddah, Saudi Arabia

ABSTRACT

Terrorism and violence are used by miscreant groups and individuals to disrupt the normal course of events. While not a new phenomenon, the information age offers new and innovative methods for spreading messages related to terrorism and expansion through recruitment on social media. These observations are alarming due to the broad reach and speed of propagation made available by social media. To ensure safety, harmony, and peace, it is important that the use of social media for terrorism is minimised. We discuss the various methods used by terrorists on social media to increase exposure and identify how the inherent structure of social media, the amount of data available, and language understanding pose challenges as well as opportunities for control efforts. We propose a strategy for restraining terrorist activities through data mining methods on big data created by social media in combination with natural language processing for language understanding and social network analysis for uncovering the underlying structure and association of terrorist groups and their activities

KEYWORDS

Big Data, Data Mining, Social Network Analysis, Natural Language Processing.

1. INTRODUCTION

There has been a war waging on between right and wrong since the dawn of time. The dictionary definition for terrorism is "the unlawful use of violence and intimidation, especially against civilians, in the pursuit of political aims," [1] which describes it as illegal and, hence, wrong by the prevailing norms. Since the commercial availability of the Internet in the late 1980s [2], users of computing devices are now connected affecting many aspects of society. While adding significant value, the Internet has opened loopholes and new forms of access for miscreants to exploit to perform their illegal efforts with greater ease. There is a dire need to curtail the use of emerging technologies for unlawful actions. New perspectives and methods have been developed with the study of counter-terrorism becoming a major research area. Scholars from social sciences, political sciences, and technological areas have joined efforts to better understand the phenomenon [3].

The Internet also gave rise to a new definition of social interaction through the extended use of social media. As social networks are less than two decades old, all earlier mediums of socialisation combined did not create such a wide audience with personalised experiences, largely due to Artificial Intelligence and Machine Learning methods. Services such as Facebook, Twitter,

David C. Wyld et al. (Eds) : AI, CSTY, NMOCT - 2018
pp. 35–42, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81404

YouTube, and Instagram have revolutionised the way people interact. These opportunities also opened doors for terrorist organisations to more efficiently recruit, grow, train, and communicate with followers, supporters, and donors. Social media is also an effective platform for spreading propaganda, ideological thoughts, and hate material [4]. Radicalisation that once required personal contact is now spread by social media through digital materials (audio, video, images, and messages). An alarming aspect is the access to local groups and societies from remote locations, which was not possible before. Social media has been driving and affecting events since its proliferation into the society, such as the international covert influencing of the 2016 U.S. presidential election [5].

Through this ability to influence, the power of social media has been used by terrorists to coordinate and execute attacks. The most effective use of social media for terrorism purposes was by Islamic State (ISIS) for spreading their ideology and propaganda materials. The distinguishing feature of their operation was the quality of the material and the ability to disseminate information through social channels [6]. Additionally, ISIS used high velocity content executed with great efficiency [7].

Terrorism-related information intended for the spread across social media targets a young audience who are within the prime recruiting age [8]. This information is also spread in multiple languages making it accessible to a broader audience [9]. Social media also enables terrorist organisations to engage directly with the public through the hijacking of the identities of credible figures. This method is performed, for example, by being the first to reply to a tweet from a notable person with a large number of followers. As the commenting thread expands, the illicit remarks gain more views. Confronting people and organisations who hold a credible reputation and enticing them into a discussion is another approach used by terrorist organisations to increase social mileage. A common strategy by insurgents is to spread their messages and drive communications through the use of disseminators located outside a war zone. In addition, algorithms developed by social services to connect people and provide suggestions based on personal preferences, history, and other information often favour terrorist organisations in spreading their messages [10]. These social algorithms differ in how they connect people. For instance, Facebook users see posts that are "personalised based on past clicks," including the 'Like' button, and on items' popularity among other users with similar preferences [11]. Google presents results based on the location and previous searches and clicks [11]. These algorithms become extremely accurate in projections over time, and miscreant groups can leverage these mechanisms for information propagation and dissemination.

Terrorist groups using social media's capabilities to disseminate information and recruit also makes them vulnerable. Social media operate through global platforms with open access, which leave digital signatures for law enforcement agencies to tracking them down. Successful operations to curb malicious activities and track those responsible have been performed with the help of open source information [10]. Analysing available information using intelligent tools and methods enables localisation and riddance of terrorists. By acting swiftly on this available information, it also becomes possible to stop incidents before they occur.

Social networks provide a graph of linkages of individual social accounts and the information contained in them. Graph analysis techniques such as cluster graph structure and graph vertex analysis [21] are used to associate people to groups and identify relationships. Terrorist organisations operate in such a way that that social networks can provide the first point of contact with potential candidates for recruitment. Once initial contact is established further communication is shifted to an encrypted channel where tracking becomes nearly impossible, which is further complicated by privacy and protection laws prevalent in various countries around the world. Thus, capitalising on this initial window of opportunity through the information available from open source social networks is of utmost importance.

In this section, we identified the role of social media in human society. We identified the vulnerabilities of social media and how it can be used by miscreant groups to gather traction, recruit followers, and spread their messages. We looked at how terrorist organisations prepare professional quality materials with high velocity and leverage learning algorithms. We identified that although social platforms offer advantages for terrorists in spreading their message, they also include vulnerabilities with traceable signatures. The mitigation goal is to identify and stop the spread of malicious content and contain it within the realm of social platforms.

The remainder of the paper is organized as follows. The next section discusses the inherent problems faced in addressing these issues. We then propose strategies that can be used to counter terrorist activities to achieve harmony and peace. This is followed with a conclusion and references.

2. THE INHERENT PROBLEM

In the information age, enormous amounts of data are created with exponential increases in recent years. With social media, huge volumes of data of different varieties are generated rapidly. Dealing with all this data is challenging as traditional methods of analysis such as guessing, constructing hypothesis and testing with data based experiments do not perform well [22] because of the sheer volume and variety of data, hence new methods uncovering the insights in data must be devised. Research in Big Data is considering new techniques [23-27] with the core challenge of how to process data to extract useful information. Computational resources seem to be second runners-up in this race with Big Data leading the way. Classic learning and intelligence methods also fail to perform well on large data set, and new "Deep Learning" techniques [23, 24] are being devised to take advantage of the available information.

Interpreting natural language comes easily to humans. However, the same is not true with computing machines, and processing natural language data is critical in uncovering terrorism-related information. This poses another challenge for the processing of Big Data produced by social media sources.

Social interactions are complex as are the relationships between entities in a social network. Complex graphs must be evaluated to extract useful information. Uncovering these relationships and making sense of them is critical to identifying terrorist groups, their recruitment strategies, and their information dissemination methods. Complex graph analysis techniques such as spectral clustering, information maps [28] must be employed for this.

Extracting meaningful information and curbing terrorist-related activities require dealing with Big Data, natural language processing and network analysis. In the next section, we propose strategies for dealing with these problems.

3. PROPOSED STRATEGY

The following strategies address the three problem domains identified with the aim of combating terrorism using the latest technology.

3.1. Data Mining on Reduced Data

Extracting information on terrorism and its allies from Big Data requires processing large amounts of data, which is sometimes not feasible due to limitations on computational resources and timing constraints. Innovative methods are required to reduce the data while preserving information content, such as the following data reduction methods.

- Dimension reduction techniques based on clustering, map-reduce implementations of existing dimension reduction methods, feature selection techniques, and fuzzy logic implementations. PCA, SVD, eigenvalue/eigenvector decompositions.

- Reduce the velocity of data streams before entering into storage (pre-processing). In a specific use case from [13], the proposed algorithms show that data reduction performs effectively, and the memory requirement is reduced from 3 TB to 300 GB of RAM.

- Data sampling [14] is useful when data sizes become too large to practically deal with the entire dataset simultaneously and has been used extensively in data mining applications. Sampling techniques include simple random sampling, stratified sampling, systematic random sampling, and cluster sampling.

- Network theory approaches are used to extract topological structures of unstructured data [15].

Using these methods, we propose to reduce the data while preserving the information content. Only the techniques that preserve the underlying semantics of the data will be helpful as we will be applying data mining techniques on the reduced dataset.

Data mining is a powerful approach for discovering valuable information by analysing data from different dimensions, categorising it, and summarising the data relationships identified in the database. Subsequently, decisions can be made or improved based on this information. In data mining solutions, algorithms can be used independently, or more than one can be applied to achieve the desired results. It can be employed using some algorithms to explore data, while other algorithms are used to extract specific data to find a specific outcome. For example, clustering algorithms, which recognise patterns in data, can be used to group data into different n-groups. Data contained in each group are considered reasonably consistent so that a decision model can be created based on the results. Multiple algorithms can be applied within one solution to perform separate tasks.

Stored data is divided into predetermined groups. The classification algorithm uses a training data set, where each record is predefined in a different class for building a learning model, and a testing dataset, which classifies and labels every record with an unknown class. Classification is sometimes called supervised learning because class labels are known in the training dataset.
Clustering, or unsupervised classification, is a method that separates data into groups of members that belong together based on some characteristic. Class labels in clustering are initially unknown, so a clustering algorithm discovers acceptable classes and assign each item to the corresponding group. Typically, clustering provides a broad view to the user of what is happening in the database. Clustering does not require prior knowledge of the groups and their members, which is useful for separating terrorist-related data from other data.

Association rules are used to discover elements that frequently co-occur in a dataset that contains multiple independent selections. The association rules approach includes two phases. The first is support, where frequent item sets are identified. The second is confidence, where conditional probabilities are identified in transactions in which items continually appear together. In the context of this article, association rules are particularly useful in extracting relationships between terrorist groups and their recruits.

Sequential patterns are anticipated from the data by mining it for patterns that appear frequently. Known patterns learned from a terrorist dataset as the one maintained by the Global Terrorism Database (GTD) [29], extracted using natural language processing methods can be useful to identify similar patterns from the data to uncover additional terrorist-related content.

Regression algorithms are useful in predicting future values of data by analysing the behaviour of data over a period. These techniques can be useful for predicting upcoming terrorist-related activities that may occur.

Once data has been reduced to a manageable size, the data mining techniques of classification, clustering, association, sequential pattern discovery, and regression can be used more effectively for identifying terrorism and extracting recruitment-related information and strategies of different groups of terrorists. With a reduced data set it is relatively easy to identify terrorism and the hidden relationships among terrorists that may result in terrorist acts. Data mining on the reduced set with clustering to gather specific data into groups can be helpful in segregating terrorist groups. Grouping terrorist data into homogeneous classes or clusters can provide a comprehensive understanding of terrorist behaviours, while predicting the likelihood of terrorist activities in a reasonable amount of time.

The process of using data mining involves the following steps:

- Establish domain (terrorism and allied activities) understanding with relevant prior knowledge and identification of end-user goals.
- Build a dataset using natural language methods, as described in the next section, by using known keywords and entities related to terrorism and terrorists.
- Pre-process data for reduction and cleanliness.
- Identify useful features in the data.
- Appropriately apply the data mining strategies of classification, clustering, association, sequential pattern discovery, and regression to understand and predict terrorist activities.
- Use data mining algorithms to search for patterns in the dataset to identify additional malicious activities.
- Extract interesting patterns that distinguish terrorist-related activities from the rest of the data.
- Document and report the observations.

3.2. Natural Language Processing

Making sense of natural language is fundamental in identifying malicious information. Natural language processing is the automatic analysis and representation of human language, which enables computers to perform a wide range of natural language-related tasks [16]. With the availability of large amounts of data, deep learning methods can be used that employ multiple processing layers to learn hierarchical representations of data. A variety of model designs and methods have blossomed in the context of natural language processing [17].

Leveraging the available language data, we can employ deep methods to extract terrorism-related information. This approach involves making semantic sense of the text and identifying underlying patterns to characterise terrorist-related information. Crawling social media sites and using natural language processing methods to analyse the content against keywords using state-of-the-art matching methods, such as distance-based matching. Identifying entities with connections to both organisations and names that have been banned and declared terrorist can be accomplished

using language processing techniques. Natural language processing tools can be used to scan the web for unwanted material and report for further analysis and processing. These methods must perform above a certain threshold for the system to work appropriately and result in as few false alarms as possible.

Natural language processing methods based on keywords and known entities are also useful for preparing and refining a continuously evolving dataset of terrorist groups, organizations, and linked people. This dataset is a prime resource for further learning and processing using Big Data analysis methods reviewed above as well social network analysis methods discussed in the following.

3.3. Social Network Analysis

With a dataset in place and a continuous stream of available data from social media, data mining methods can extract useful information. However, this processing can be further improved using social network analysis techniques to unravel any underlying structures, relationships, and associations. Social network analysis is a collection of techniques that support statistical investigations on the patterns of communication between groups. Social scientists use these to analyse connected groups, and they form a basis of techniques for situational awareness and decision making in law enforcement applications [18].

A linkage map of terrorist organisations can be created using social network analysis [12] from which a frequency of co-occurrence of names of organisations can be used as a basis for inferring the intensity of the links. Concepts from graph theory play a pivotal role in network analysis, such as how an adjacency matrix will reflect the closeness of organisations. In addition, graph-theoretic concepts of centrality and between-ness provide further insight into the operation and structure of terrorist organisations. Using nodal analysis in social networks, a terrorist group can be rendered impotent by identifying and targeting their nodal or key points.

Social network analysis assesses the examined social aspects based on structures, which incorporate group members represented by nodes and their interconnections. As opposed to other quantitative strategies that centre around the portrayal and total investigation of the qualities of the actors who make up the exploration populace, social network analysis expects that to understand the social phenomenon, it is helpful to guide out and break down the arrangement of ties among the actors and the manners by which these social patterns shape actions. Thus, this approach can provide information on the decision making, group dynamics, and the outcomes of collective actions. The methodology for studying violent groups is broken down in the following, as suggested by [3]:

- Mapping the group with characteristics of parallel ties, symmetric/asymmetric, negative/positive, and the quality of ties, including measurement of time spent together, recurrence of communication, and size of associations.

- Division of power within a group, including progressive gatherings, actors having greater number of ties, thick structures or a heap of associated subgroups situated in vital areas. Proportion of status or centrality measures, idea of impact as an element of actor's significance.

- Structure and subgroups, including levels of cohesion and degree hierarchies, balance between efficiency with either a low number of repetitive connections or a high level of group centrality.

- Robustness and survivability with high density and a large number of redundant ties.

This framework leads to a deep analysis of terrorist groups and their modus operandi. Implementation of this is made possible using tools such as NetworkX [19] and SNAP [20].

4. CONCLUSION

In this paper, we presented the methods used by terrorists to spread their messages using social media. It is understood that containing terrorism-related material on social media is critical. We analysed the associated problems and proposed strategies towards a solution for containment of terrorism-related activities. The proposed strategy includes the use of natural language processing to build and expand a dataset by looking for terrorist related data on social networks, the reduction of the data using data sampling techniques, the use of data mining methods on reduced data to identify the patterns and extract useful information and the use of social network analysis to uncover the associations and relationship between individuals and terrorist groups, their structure and their modes of operations.

REFERENCES

[1] Terrorism definition, Available at https://en.oxforddictionaries.com/definition/terrorism

[2] History of Internet, Available at https://en.wikipedia.org/wiki/History_of_the_Internet

[3] P. Arie and P. Ami, "Social Network Analysis in the Study of Terrorism and Political Violence" Southern Illinois University Carbondale, OpenSIUC Working Papers 2010

[4] Md. S. Hossain, "Social Media and Terrorism: Threats and Challenges to the Modern Era" South Asian Survey, Vol 22 (2), pp. 136-155, 2018

[5] B. Bender, "SOCIAL MEDIA AND POLITICS: THE 2016 US PRESIDENTIAL ELECTION", BrandBa, 2017

[6] Alexander Meleagrou-Hitchens, et al., The Travelers: American Jihadists in Syria and Iraq, George Washington University Program on Extremism, February 2018, 33. Available at: https://extremism.gwu.edu/events/travelers-american-jihadists-syria-and-iraq

[7] Charlie Winter, "Fishing and ultraviolence: So-called Islamic State is known for its brutality. But it's also hooking people in far subtler ways," BBC, August 1, 2015. Available at: http://www.bbc.co.uk/news/resources/idt-88492697-b674-4c69-8426-3edd17b7daed

[8] Aris Roussinos, "Jihad Selfies: These British Extremists in Syria Love Social Media," Vice, December 5, 2013. Available at: https://www.vice.com/en_us/article/gq8g5b/syrian-jihadist-selfies-tell-us-a-lot-about-their-war.

[9] A. Alexander, Digital Decay: Tracing Change Over Time Among English-Language Islamic State Sympathizers on Twitter, George Washington University Program on Extremism, October 2017

[10] G. Ratnam, B. Misztal, S. Hughes, J. Geltzer, R. L. Strayer, "Digital Counterterrorism: Fighting Jihadists Online", Report Bipartisan Policy Center, 2018

[11] M. El-Bermawy, "Your Filter Bubble is Destroying Democracy," Wired, November 18, 2016. Available at: https://www.wired.com/2016/11/filter-bubble-destroying-democracy/.

[12] B. Aparna. "Social network analysis of terrorist organizations in India", Institute for Defence Studies and Analysis 2005.

[13] H. Bronnimann, B. Chen, M. Dash, P.J. Haas, and P. Scheuermann, "Efficient Data Reduction with EASE," Proc. ACM SIGKDD, 2003.

[14] Weinstein M et al (2013) Analyzing big data with dynamic quantum clustering. arXiv preprint arXiv:1310.2700.

[15] M. Trovati "Reduced topologically real-world networks: a big-data approach" Int J Distrib Syst Technol (IJDST) 6(2):13–27

[16] E. Cambria and B. White, "Jumping NLP curves: A review of natural language processing research," IEEE Computational Intelligence Magazine, vol. 9, no. 2, pp. 48–57, 2014.

[17] T. Young, D. Hazarika, S. Poria,& E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]". IEEE Computational Intelligence Magazine, 13, 55-75, 2018

[18] P. Svenson, P. Svensson, H. Tullberg, F. Ledningssystem, F. Ledningssystem, F. Ledningssystem, "Social Network Analysis And Information Fusion For AntiTerrorism", In Proc. CIMI, 2006

[19] H. Aric, S. Pieter, & S Chult, Daniel. "Exploring network structure, dynamics, and function using networkx". United States, 2008.

[20] J. Leskovec and R. Sosic, "SNAP: A General Purpose Network Analysis and Graph Mining Library", CoRR, 2016

[21] Q. D. Truong, T. Dkaki, Q. B. Truong, "Graph Methods for Social Network Analysis", 2nd EAI International Conference on Nature of Computation and Communication – ICTCC, pp-276-286, 2016

[22] R. Kitchin, "Big Data, new epistemologies and paradigm shifts", Big Data & Society, Sage Publications, April, 2014.

[23] J. Cała, P. Missier, "Selective and Recurring Re-computation of Big Data Analytics Tasks: Insights from a Genomics Case Study", Big Data Research, Vol 13, pp 76-94, 2018

[24] H. Estiri, B. A. Omran, S. N. Murphy, "kluster: An Efficient Scalable Procedure for Approximating the Number of Clusters in Unsupervised Learning", Big Data Research, Vol 13, pp 38-51, 2018

[25] D. Lakshmipadmaja, B. Vishnuvardhan, "Classification Performance Improvement Using Random Subset Feature Selection Algorithm for Data Mining", Big Data Research, Vol 12, pp 1-12, 2018

[26] K., Savita. "Impact of big data and social media on society". Global Journal for reseach Analysis. Vol 5, pp 437-438, 2016

[27] C. Debas, "Big data analytics for exploratory social network analysis", International Journal of Information Technology and Management, Vol 16(4), 2017

[28] M. William, C. Campbell, K. Dagli, and C. J. Weinstein, "Social Network Analysis with Content and Graphs", Lincoln Laboratory Journal, Vol 20(1), 2013

[29] National Consortium for the Study of Terrorism and Responses to Terrorism (START). (2018). Global Terrorism Database [Data file]. Retrieved from https://www.start.umd.edu/gtd

ACCESS NETWORK IMPROVEMENT FOR A WLAN BASED ON 802.1X AND CAPSMAN PROTOCOLS

Fabián Cuzme-Rodriguez[1], Carlos Pupiales-Yépez[1], Mauricio Dominguez-Limaico[1], Carlos Bosmediano-Cárdenas[1] and Walter Zambrano-Romero[2]

[1]Universidad Técnica del Norte, Facultad de Ingeniería en Ciencias Aplicadas, Ibarra-Ecuador
[2]Universidad Técnica de Manabí, Portoviejo-Ecuador

ABSTRACT

This article describes the implementation of a method to access to a wireless network based on Radius with Mikrotik equipment. It is applied EAP-TTLS authentication based on open source software to provide AAA services and a LDAP directory to save user's accounts. This approach allows a better access control and distribution of network resources. As a plus, the work includes the implementation of Mikrotik's CAPSMAN protocol which enablesa centralized control of all access points (AP) emitting the same SSID. This proposal improves the performance of the network and user´s experience.

KEYWORDS

Radius, Mikrotik, EAP-TTLS, 802.1x

1. INTRODUCTION

Since wireless devices showed up, network designers have developed different approaches to guarantee those devices can have access to the services offered by a wireless network. The access to the network is done by a wireless access point which establish centralise and manage the working rules applied in the network. Contrary to a LAN, a WLAN offers several advantages to users and network managers such as nomadicity, deployment reduction costs, and scalability; however, the main disadvantage of wireless networks is the data vulnerability due to RF waves can be detected by anyone at any moment inside the coverage zone. Therefore, it is mandatory to implement access policies to avoid eavesdroppers can steal valuable information of our networks. Currently is common to find wireless networks in public and private spaces that offers free access to the Internet where users enter their personal information; for this reason, it is significant to implement robust and mores trusted access mechanisms such as Radius or Kerberos.

Several works show that wireless networks are unsafe thus it is necessary to implement security approaches to avoid informatics attacks or in the worst case, reduce the effects of those attacks. [1] Proposes the implementation of an open code server, DIAMETER EAP, for authentication and authorization of remote users from the networks of a large corporation. Additionally, [2] states that information security plays an important role in any network because it guarantees the integrity and confidentially of data; in fact, the 802.1x protocol gives the network the ability to

David C. Wyld et al. (Eds) : AI, CSTY, NMOCT - 2018
pp. 43–49, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81405

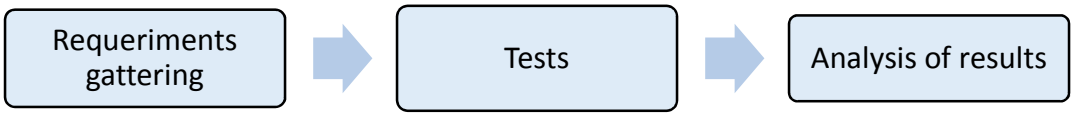
control the access based on ports due to all traffic is denied if users have not been authenticated in the authentication server first.

There are technological solutions implemented in government agencies that try to solve both security and access issues in a wireless network; for instance, [3] has developed a solution based on CAPsCAN and RouterOS which substantially improves the access and security aspects of the networks. Moreover, [4] uses CAPsCAN, WDS, and NDLC to solve similar issues presented in [3], but in this case the offered solutions is implemented in a university campus. Even though network managers implement security policies in wireless networks, the network itself might be still vulnerable due to new and sophisticated methods and tools used in ethical hacking; therefore, it obligates to create stronger and dedicated authentication mechanism that fit to the application and service where it will work. For example, [4] points out that in the authentication mechanism RADIUS the access points are authenticated by a shared static key which is not suitable for wireless networks; for this reason, it is possible to use the approach TPM, Trusted Platform Module, besides the authentication by RADIUS only.

This work is focused on improving the performance and user's experience of FICA's WiFi network which has approximately 2000 users distributed into students, teachers, and administrative personnel who access to the offered services by a simple Hotspot system as the only management tool used to discriminate against traffic, priorities, and bandwidth. This approach is valid for low traffic patterns; however, the number of electronic devices and users that try to access to the network is increasing constantly which cause that the simple Hotspot approach becomes inefficient because of the equipment's overheat, service unavailability in peak hours, 10 am – 1 pm, and a very low performance of the network all day long. Therefore, we propose a scenario where the access to the network is based on a Radius server, 802.1x protocol, and a simple queues to allow users to access to the network with their own credential that makes possible to control the data rate users obtain and can reach.This paper is structured as follows. Section 2 describes briefly the problem this works intents to solve. Section 3 specifies the approach and tools used to improve the performance of the network. Additionally, section 4 shows the results obtained before and after the implementation of the proposed approach and finally section 5 concludes the work with recommendations and future work.

2. PROPOSAL AND METHODOLOGY

This work uses an exploratory and analytic methodology where we explore new solutions to solve the problem of access control that a real network has. We follow the following process:



To address the inefficiency of access control and low performance of the network, this work proposes to implement the 802.1x protocol and the authentication mechanism EAP-TTLS in a centralized server which will be attached to every single access point, of the brand Mikrotik, in the network. The implementation considers the following features:

- The devices allowed to connect to the networks are laptops, tablets, and smartphones.
- Authenticator devices, APs, must support the management system CAPsCAN, RouterOS, and 802.1x protocol.

- The authentication server will be developed entirely in an open source software.

It's important to remark that we applied a combined protocol which consist of two stages. The first one establish the TLS tunnel with security in the transport layer. The last one, consist of encapsulate the TLS connexion using the EAP method. This approach reduces the system complexity since the radius server requires only one digital certificate to authenticate the user instead of sending several certificates to all the devices attached to the network.

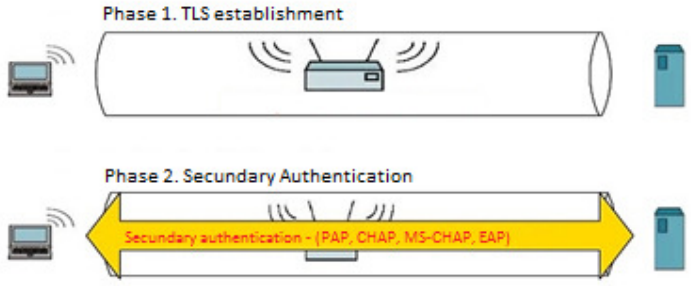


Figure 1. EAP-TTLS Method

2.1. Requirements

2.1.1. SUPPLICANT

The devices allowed to connect to the network must support as operative system at least Windows 8, any distribution of Linux, and Android 1.6. IOS does not need additional software to perform the authentication. For previous and different versions of Windows and Linux it is mandatory an additional software that supports EAP-TTLS. Table 1 specifies those requirements.

Table 1. Technical requirements for Supplicant.

Operative System	EAP-TTLS
Windows 7	SecureW2
Windows 8 / 8.1	Native Client/ SecureW2
Windows 10	Native Client
Ubuntu/Debian/Centos	Native Client
Android OS/ IOS (Iphone-OS)	Native Client

2.1.2. Authenticator (Access Points)

The access points involved in the solution are Mikrotik model CAP-2n and routerOS RB1100. These were chosen because they offer a simple adaptation to a centralized management model such as CAP mode, because routerOS gets along with 802.1x and CAPsCAN mode.

2.1.3. Authentication server

In the authentication server will be running the OS Debian 8, the application FREERADIUS, and the active directory LDAP. The server should has as technical features to respond the requests from all users at least a hard-disk with 10 GB free for storage, processor' speed of 2 GHz, and 1GB in RAM memory. The chosen equipment counts with 3.5 GHz in the processor, 2 GB as RAM memory, 100 GB free in the hard disk, and a Gigabit Ethernet interface.

2.2. Physical and Logical Topology

The initial and the final scenarios have almost the same topology; however, the difference in each scenario is how equipment are connected and configured.

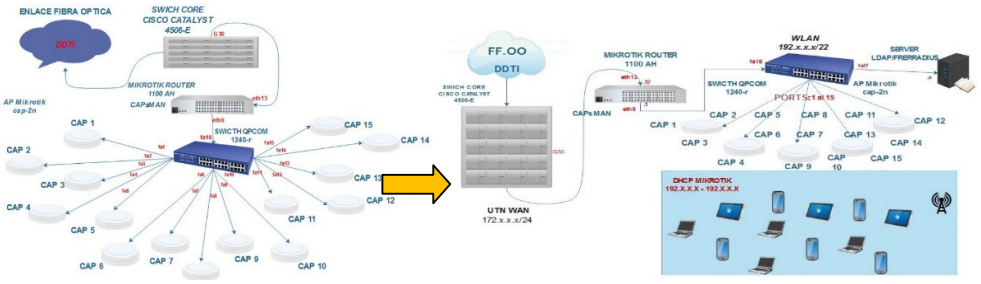


Figure 2. Topologies before and after implementation

2.3. Authentication Server

- **FreeRADIUS:** All dependencies and packets can be extracted from their own data repository using the procedure showed in figure 2. The advantage of using Debian 8 – Jessie as primary operative system is the it is available for both 32 bits and 64 bits architectures which reduces time at the moment of installation and configuration of any software and their dependencies.

- **OPENLDAP:** FICA's WiFi does not have user's directory to save the access credentials used in the authentication process, thus we create a directory using OpenLDAP. The structure of the data base implemented in the network "ficawifi" is showed in figure 3.

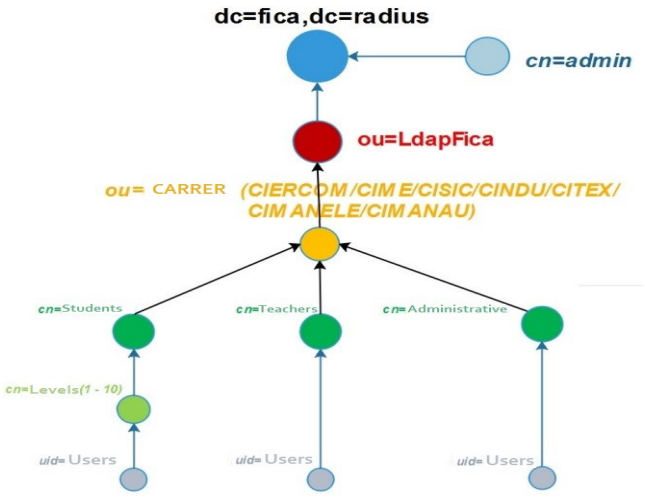


Figure 3. Hierarchical structure LDAP - FICA

Once the service LDAP has been installed, it is necessary to add certain information to its main file by using the command ***dpkg-reconfigure slapd.*** The information filled in the file is:

- Domain name: utn
- Organization name: fica.radius
- Password: ****
- Database engine: MDB

Additionally it is mandatory to enable de purge of ***slapd*** packets and avoid ***LDAPv2*** protocol.

2.4. Authenticator (CAP andCAPsMAN)

There are 15 APs available in the wireless network which must have the same SSID, so every AP is configured in CAP mode. This allows the routerOSMikrotikto manage the APs by CAPsMAN. All APs should be in the same network domain so that every AP can communicate to each other. This can be done configuring a fixed-IP or with a VLAN in CAP mode. Additionally, the mode CAPsMAN must be enabled to have a centralized management of the network. This configuration is presented in figure 4.

Name	SSID	Channel	Datapath	Security
cfg-AP1	ficawifi	channel1	datapath1	RadiusPASS
cfg-AP2	ficawifi	channel11	datapath1	RadiusPASS
cfg-AP3	ficawifi	channel6	datapath1	RadiusPASS
cfg-AP4	ficawifi	channel11	datapath1	RadiusPASS
cfg-AP5	ficawifi	channel6	datapath1	RadiusPASS
cfg-AP6	ficawifi	channel9	datapath1	RadiusPASS
cfg-AP7	ficawifi	channel1	datapath1	RadiusPASS
cfg-AP8	ficawifi	channel6	datapath1	RadiusPASS
cfg-AP9	ficawifi	channel11	datapath1	RadiusPASS
cfg-AP10	ficawifi	channel6	datapath1	RadiusPASS

Figure 4. Configuration parameters for every CAP

According to a first analysis, the occupancy of the network is distributed in 80% for students and 20% for teachers and administrative personnel. This initial analysis would state that students need more data rate than teachers; however, both students and teachers should have the same privileges in terms of data rate and connection. The easiest way to limit the data rate is using simple queues; for this reason, we create two queues for teachers and students.

3. RESULTS

The success of the approach implemented in this work is measured in terms of user's perception and data rate control.

3.1. Perception of the user

To know how users react to the changes in the network, we use the simple approach of interviews. The sample for the analysis is 66 users who are interviewed before and after the implementation of the proposed solution. The period of analysis is from March to August of 2017 and the analysed points are:

- What was the user's experience regarding service unavailability, network coverage, and web browsing?

- Which were the uplink and downlink data rates?

- Conclusion after interviews show that for users the parameters network performance and data rates improved significantly with the implementation of a better network access approach. The results are presented in table 2.

Table 2. User's perception before and after implementation

Parameter	Hotspot			Radius		
	B	G	E	B	G	E
Network Performance	67%	25%	8%	22%	40%	38%
Data rates	57%	38%	5%	10%	54%	36%
B: Bad **G: Good** **E: Excellent**						

3.2. Data Rate Control

In the first stage of the study, between February and March 2017, it is noticeable the inefficient use of resource due to there was not a control for assign data rates. Users used to be able to connect several devices with the same credentials. The effects of this problem was reflected as an uncontrolled assignment of bandwidth for present and future users; in fact, a single user used to use a large proportion of channel bandwidth while others used to be assigned with just few kbps for navigation.

After applying the Simple Queues approach in the Mikrotik router we got excellent results in the period July – August 2017 that can be synthetized as follow:

- The data rate assigned for each user was 6 Mbps in July and the first days of August instead of the 13 Mbps usedin the first stage of the study. Now the average number of users are 150 who are assigned with the bandwidth they request and which is enough for their application thus all users can access to the network and traffic patterns are distributed in a better way.

It is significant to remark that even though the data rate itself decreases, the efficiency in the assignment of it increases being more than enough 6 Mbps. Figures 5 and 6 show the traffic pattern before and after the implementation of simple queues approach, 802.1x, and network segmentation.

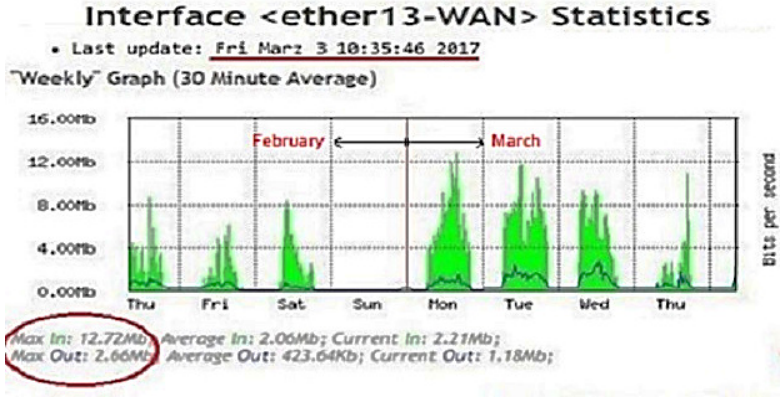


Figure 5. Traffic pattern before implementation

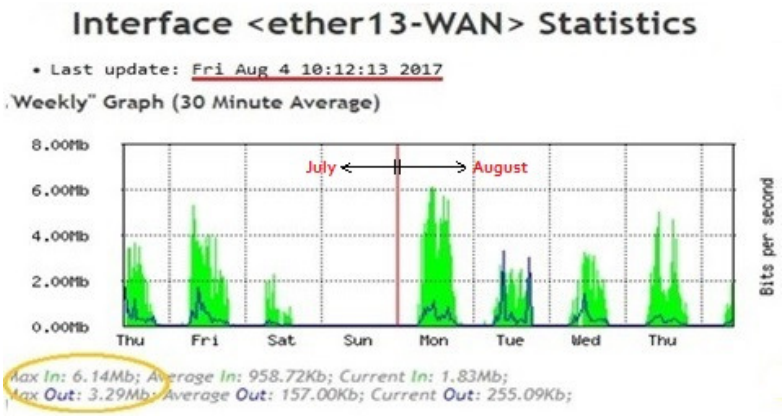


Figure 6. Traffic pattern after implementation

4. CONCLUSIONS

With the implementation of the proposed approach was possible to reach a network performance of 78% comparing to the 34% of performance gotten with the access based on a simple hotspot. Additionally, now data rates for downlink and uplink are assigned more efficiently since users do not share the same credential which used to congest the network. As a consequence, more users can have access to the services offered by the network.

ACKNOWLEDGEMENTS

We would like to thank to Universidad Tecnica del Norte from Ecuador for the strong support for the development of this article.

REFERENCES

[1] Wu, W. T., Chen, J. C., Chen, K. H., & Fan, K. P. (2015, June). Design and implementation of WIRE Diameter. In Information Technology: Research and Education, 2015. ITRE 2015. 3rd International Conference on (pp. 428-433). IEEE.

[2] Qian, Q., Li, C., & Zhang, X. (2013, August). On Authentication System Based on 802.1 X Protocol in LAN. In Internet Technology and Applications, 2013 International Conference on (pp. 1-4). IEEE.

[3] García V. R., (2018). Análisis de implementación de una red CAPsMANMicroTik en el Gobierno Autónomo Descentralizado Provincial de los Ríos. Tesis de pregrado.

[4] Santi DwiRatnasari, E. F., (2017), Implementación de CAPsCAN y Sistema de Distribución Inalámbrica WDS en SMK Integrado al ISHLAHIYAH SINGOSARI MALANG.

AUTHOR INDEX