A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

* "International Journal of Computers and Their Applications is Peer Reviewed".

Volume 29, No. 1, March 2022 ISSN 1076-5204

International Journal of Computers and Their Applications

A publication of the International Society for Computers and Their Applications

ISCA Headquaters • 278 Mankato Ave, #220, Winona, Minnesota 55987, USA
Phone 507-458-4517 • Email isca@isca-hq.org • URL: http://www.isca-hq.org

Editorial
March Issue 2022

It is my distinct honor, pleasure, and privilege to serve as the new Editor-in-Chief of the International Journal of Computers and Their Applications (IJCA). I have a special passion for the International Society for Computers and their Applications. I have been a member of our society since 2014 and have served in various capacities. These have ranged from being on program committees of our conferences to being Program Chair of CATA 2021 and CATA 2022 and currently serving as one of the Ex-Officio Board Members. I am very grateful to the ISCA Board of Directors for giving me this opportunity to serve society and the journal in this role.

I want to extend my special thanks to Dr. Wenying Feng, the outgoing Editor-in-Chief of IJCA, and the past editors-in-chief for contributing so much to IJCA and providing me with a firm foundation to move forward. I would also like to thank all the Associate Editors, the editorial staff, and the authors for their valuable contributions to the journal. Without everyone's help, the success of the journal would be impossible.

I look forward to working with everyone in the coming years to maintain and further improve the journal's quality. I want to invite you to submit your quality work to the journal for consideration of publication. I also welcome proposals for special issues of the journal. If you have any suggestions to improve the journal, please feel free to contact me.

Dr. Ajay Bandi
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, MO 64468
Email: AJAY@nwmissouri.edu

In 2022, we have four issues planned (March, June, September, and December). March issue includes the best papers from the SEDE 2021. Drs. Fred Harris, Rui Wu, and Alex Redei are the guest editors of this March issue. June issue will include the selected best papers from CAINE 2021. The third issue in September will contain the best papers from CATA 2022. The last issue is taking shape with a collection of submitted papers.

I would also like to announce that I will begin searching for a few Associate Editors to add to our team. There are a few areas in which we would like to strengthen our board. If you would like to be considered, please contact me via email with a cover letter and a copy of your CV.

Ajay Bandi, Editor-in-Chief
Email: AJAY@nwmissouri.edu

Guest Editorial:

Special Issue from ISCA Fall--2021 SEDE Conference

This special issue of the International Journal of Computers and their Applications (IJCA) is a collection of four refereed papers selected from SEDE 2021: the 30th International Conference on Software Engineering on Data Engineering, held October 11-12, 2021. This conference, due to the pandemic, was held virtually.

Each paper submitted to the conference was reviewed by at least two members of the international program committee, as well as by additional reviewers, judging for originality, technical contribution, significance and quality of presentation. The proceedings for this conference can be found online at https://easychair.org/publications/volume/SEDE_2021. After the conference, the six best papers were recommended by the program committee members to be considered for publication in this special issue of IJCA. The authors were invited to submit a revised version of their papers. After extensive revisions and a second round of review, these papers were accepted for publication in this issue of the journal.

The papers in this special issue cover a broad range of research interests in the community of computers and their applications. The topics and main contributions of the papers are briefly summarized below.

ISLAM KHALIL, SHERIF EL-KASSAS, and KARIM SOBH of The American University in Cairo, Cairo, Egypt present their paper "A Multi-Modal, Pluggable Transaction Tamper Evident Database Architecture." In this paper they present the architecture for a multi-modal tamper detection solution with a primary goal of being easily retrofittable into existing systems with minimal intervention required from system developers or system administrators in large organizations. Their focus in this work is append-only databases like financial transactions, auditing systems, as well as technical system logs. They also pay attention to data confidentiality and leverage designs like chains of record hashes to achieve the target solution. They illustrate different ways of integrating DBKnot into existing architecture, and then go through how to leverage existing web-service configuration and definition standards to increase the seamlessness and ease of retrofitting into existing applications by automatically detecting and learning about the target web-service semantics without much need for manual human intervention.

JONATHON HEWITT, DANIEL HALL, CHRISTOPHER PARKS, PAYTON KNOCH, SERGIU M. DASCALU, DEVRIN LEE, NIKKOLAS J. IRWIN, and FREDERICK C. HARRIS, JR. of the University of Nevada, Reno present their paper "Design and Implementation of VS-TAP: The Veteran Services Tracking and Analytics Program." In this work they present VS-TAP, the The Veteran Services Tracking and Analytics Program. VS-TAP is a data gathering and analytics application with the goal of collecting, storing, and combining data from several sources into a single usable database. The web application also tracks the rate and duration of visitors that attend veteran centers and events. The program combines all the data collected from various sources that can be queried for data visualization purposes.

FENG YU of Yougstown State Univeristy in Youngstown Ohio, SEMIH CAL of Texas Tech University in Lubbock Texas, EN CHENG of the University of Akron in Akron Ohio, LUCY KERNS of Yougstown State Univeristy in Youngstown Ohio, and WEIDONG XIONG of Cleveland State University in Cleveland Ohio present their paper "Non-parametric Error Estimation for σ -AQP using Optimized Bootstrap Sampling." In this work, they employ a non-parametric statistical method, called bootstrap sampling, to assess errors of an Approximate Query Processing (or AQP) system for selections queries (or σ -AQP). They implement a prototype AQP system integrated with a bootstrap sampling engine that can estimate the standard deviation and produce confidence intervals for selection query estimations. Extensive experiments operating the prototype system demonstrated that the confidence intervals generated can cover the ground truth query results with high accuracy and low computing costs. In addition, they introduce optimization strategies for bootstrap sampling which can improve the overall computing efficiency of the prototype AQP system.

LIN HALL, PING WANG, GRAYSON BLANKENSHIP, EMMANUEL ZENIL LOPEZ, CHRIS CASTRO, ZHEN ZHU, and RUI WU of East Carolina University present their work "VR Tracker Location and Rotation Predictions using HTC Vive Tracking System and Gradient Boosting Regressor." They proposed a framework which integrates VR with machine learning to track, predict and visualize the position and orientation of VR trackers. The framework includes prediction of time series data obtained from the simulated human spine, for which they use a gradient boosting regressor model. The simulated human spine is visualized in VR. They propose ther framework can support other medical visualization applications as well.

As guest editors, we would like to express our deepest appreciation to the authors and the program committee members of the conference these papers were selected from.

We hope you will enjoy this special issue of the IJCA and we look forward to seeing you at a future ISCA conference. More information about ISCA society can be found at http://www.isca-hq.org

Guest Editors:

Frederick C. Harris, Jr, University of Nevada, Reno, USA, SEDE 2021 Conference Chair
Rui Wu, East Carolina University, Greenville, NC, USA, SEDE 2021 Program Chair
Alex Redei, Central Michigan University, Mount Pleasant, MI, USA, SEDE 2021 Program Chair

March 2022

A Multi-Modal, Pluggable Transaction Tamper Evident Database Architecture

Islam Khalil*, Sherif El-Kassas*, and Karim Sobh*
The American University in Cairo, Cairo, Egypt

Abstract

Fraud and data tampering is one of the key security risks of computer systems in general and in particular, sophisticated architecture that involves a wide array of heavily interdependent systems that communicate data using microservices, as well as simple normal user-facing systems.

The evolving risks of security threats as well as regulatory compliance are important driving forces for achieving better integrity and detecting any possible data tampering by either internal or external malicious perpetrators. We present the architecture for a multi-modal tamper detection solution with a primary goal of being easily retrofittable into existing systems with minimal intervention required from system developers or system administrators in large organizations. Our focus in this work is append-only databases like financial transactions, auditing systems, as well as technical system logs. We also pay attention to data confidentiality by making sure that the data never leaves the organization's premises. We leverage designs like chains of record hashes to achieve the target solution. After illustrating different ways of integrating DBKnot into existing architecture, we then go through how to leverage existing web service configuration and definition standards to increase the seamlessness and ease of retrofitting into existing applications by automatically detecting and learning about the target web service semantics without much need for manual human intervention.

Key Words: Database, security, tamper evident, chaining, lock-chain, and hash chaining.

1 Introduction

With the increasing use and ubiquity and multiple ways to use and access data across systems and as system architectures get more sophisticated and their interdependence is increasing while the range of technologies being used is widening, the need for more security and detecting fraud also increases. We propose a novel solution to protecting database integrity by providing a transparent and seamless middleware for securing database transactions against possible tampering by individuals who have full administrative access to the database and all its related infrastructure.

Such systems manage information like bank transactions, medical information, government records, as well as other critical information. Such systems often fall prey to perpetrators who are insiders or collude with insiders to commit their fraud crimes. External malicious actors are in many cases the players responsible for committing fraud and tampering with sensitive databases. Many cases involve tampering with existing systems and making fraudulent transactions that go unnoticed because they are committed by insiders who already have access and permission to the systems they tamper with.

According to the Association of Certified Fraud Examiners (ACFE) 2018 report [26], $7 Billion of losses were incurred due to internal fraud alone with an average fraud scheme going for 16 months unnoticed. Small businesses lose twice as much as big organizations due to their lack of proper access to a) Internal control processes that mitigate against such fraud and b) Systems in place that protect against such tampering.

According to Harvard Business Review [31], more than 80 million insider security breaches occur every year costing tens of billions of dollars in the US alone. In one incident $350,000 was stolen from 4 Citibank customers by employees of a software and service company that Citibank had contracted [31]. According to Accenture [6] and The World Economic Forum (WEF) [35], the cost of insider malicious activity constitutes 15% of all cybercrime. The IETF's RFC 4810 [28, 39] guidelines for "Long Term Archive Services Requirements" indicate that non-repudiation and integrity are important to any store of data to protect against potential tampering. The number of internal fraud cases resulting in compromising the integrity of organizations' data is increasing year after year [31]. For example, in the year 2010 alone, internal fraud has increased at a rate of 20%.

One of the causes of such an increase is the broadening complexity and use of IT solutions and its corresponding increase in the number of internal and external stakeholders needed to operate such systems.

Various governments have put in place different regulations to reduce/eliminate such risks. Among such regulations are the Gramm-Leach-Bliley Act by the US Federal Trade Commission (FTC) [11] which mandates that companies engaging in financial services put in place necessary measures to safe-guard their sensitive data against tampering. Another act that was decreed by the US congress is the Sarbanes-Oxley Act [22] (SOX) which mandates that companies protect their data and ensure that destruction of evidence does not occur for the purpose of later investigation of corruption and fraud cases.

* Emails: {ikhalil, sherif, kmsobh}@aucegypt.edu.

This act was made as a reaction to a number of major corruption scandals including Enron and WorldCom. The Health Insurance Portability and Accountability Act [18] (HIPAA) by the US Department of Health and Human Services (HHS) is also an example which regulates access and changes to medical records.

The goal of this work is to design a solution that enables systems based on traditional databases to be tamper-evident. Different integration models are to be discussed (on the ORM level, database level, or web service level). The primary goal is to eliminate the need for trust inside the organization while minimizing the overhead added by the solution. Ease of integration is key while requiring zero or little changes to existing systems. The solution should be able to detect tampering either by external hackers or by internal malicious employees, staff, and system administrators who have full permissions on the target database. This is done by relying more on information accountability rather than information restriction [24, 126, 129].

In the process of coming up with such a solution, a number of different technologies are examined, in addition to related work.

Example append-only applications that could benefit from our proposed solutions are server security logs, banking transactions, accounting ledgers in enterprises, notary and real-estate records, birth and death records, time and attendance systems, and many others.

Possible tampering could be committed on different levels. On a system administrator level, however the risk is that a) The sysadmin can commit the fraud and b) The sysadmin can cover-up any traces or logs of the fraudulent activity they have performed since he/she is the one responsible for all system permissions, logs, monitoring, etc.

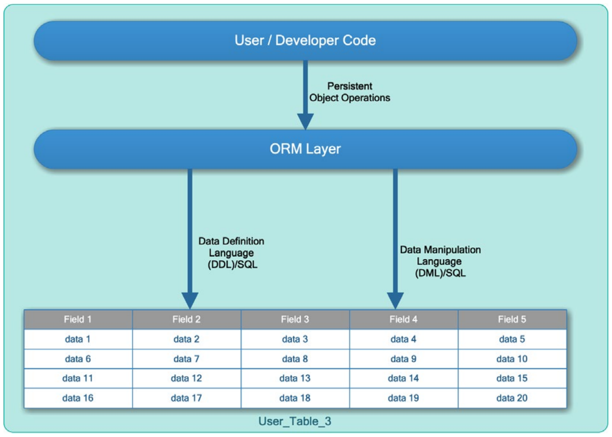
We start this paper by giving a background on some of the technologies used, then we briefly mention different related approaches to the same problem and a comparison of their corresponding features. Afterwards we go through our proposed solution, then we show some results of our experimentation followed by a conclusion. This paper is a continuation of the work done in [13].

2 Background

2.1 Object Relational Mapping (ORM)

Object Relational Mapping (ORM) frameworks [16–17, 36, 38] sit between developer applications and databases. They provide developers with full object-oriented semantics to the database allowing developers to use object oriented design to model their data without having to worry about how this maps to the database. ORM frameworks in turn take care of the mapping between data objects on one hand, and tables and relations on the other hand during database creation and definition, transactions, as well as querying.

Figure 1: Standard ORM operations shows how the ORM layer sits between the developer code and the database itself and abstracts away all of the DBMS specific relational database operations.


User / Developer Code
Persistent Object Operations
ORM Layer
Data Definition Language (DDL)/SQL
Data Manipulation Language (DML)/SQL

Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

Figure 1: Standard ORM operations

2.2 Web Services

Web services provide a standard mechanism of integrating different software systems or subsystems while abstracting away all implementation details and technologies. Web services usually provide the functionality to make database transactions as well as queries through formats like the REST API [37].

Figure 2: REST API request and response is an example of how web services work.

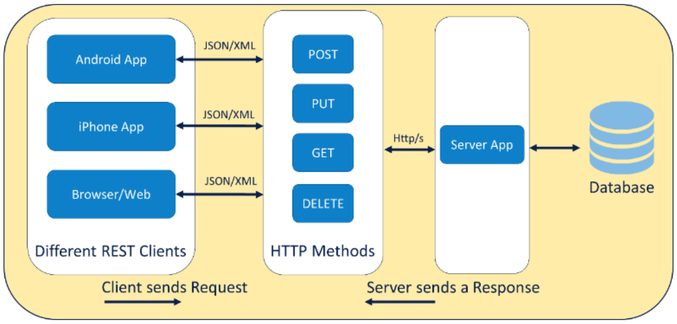


Figure 2: REST API request and response

2.2.1 REST. The definition of REST according to [27] is "Representational State Transfer".

REST is defined to be a standardized HTTP based communication scheme for systems to invoke web services across hybrid technologies without relying on any technology specific integration and thus, decouple implementations from internal technologies.

REST depends on standard HTTP methods (GET, POST, HEAD, DELETE, etc.) and uses standard HTTP return codes to communicate meaningful responses.

Contents of a REST message are usually written in formats like JSON (a javascript notation representation of data), but also other formats could be used like XML and YAML.

2.2.2 CRUD. CRUD (Create, Read, Update, Delete) are standard database operations. They are however often mapped very closely to REST API calls [32] (REST APIs have many other non-CRUD uses as well). The concept of CRUD was coined long ago before web services APIs were used and is very database specific.

2.2.3 Scenarios of REST and CRUD Mapping. With the creation of REST, there started to be many use cases that tend to show semantic similarities between parts of the two concepts.

3 Related Work

A number of different solutions have been proposed to target the problem we are addressing. Solutions vary in the way the problem is tackled. Some of them use a similar technique of chained hashes. All the solutions surveyed failed to provide a seamless and non-invasive way to get retrofitted into existing solutions with little or zero changes necessary. Another important difference is the requirement that data does not leave the users' premises.

DRAGOON [1, 23-24] is an information accountability system that relies on continuous cryptographic hashing of transactions. DRAGOON primarily relies on an external "Digital Notarization Service" rather than just a simple external transaction signer.

Amazon Quantum Ledger Database (QLDB) [4], a blockchain based database, solves part of the problem addressed in our work.

QLDB provides the ledger database service based on the premise that there is a "central" and "trusted" authority which in this case is Amazon. In this case Amazon provides the signing and trust service as well as the hosting of the actual data. Which is exactly the model we are trying to avoid and solve. Having both the storage of the data as well as the verifiability of its integrity in the hands of the same party. The difference though is that it requires data to be stored at Amazon premises meaning that Amazon needs to be depended on as a trusted host of the data.

BigchainDB [12] leverages a blockchain network to provide decentralized and an immutable database. However, due to its sophisticated setup, it does not allow seamless retrofitting into existing systems.

There are other research work like [24] that focus on documents rather than data. Some of which are designed to track documents provenance throughout their lifecycle.

Several other research works have catered to a similar problem in the domains of operating systems and file systems. Examples are [5, 8, 10, 14-15, 20, 30, 34]. But most of them either depend on a local trusted administrator or use mechanisms that require data to be moved to outside the local premises.

Summary of Related Work Comparison:

By looking at the related work, the primary gaps that our solution fills are:

- **Trust of an Insider:** Many of the solutions provide measures to protect or detect data tampering on an application level or on a database level with all requirements present in-house and within the control of the internal DBA team. This comes with the implicit assumption that the internal top-most system administrators with the highest level of access to systems and databases are fully trusted and cannot be malicious or even collude to tamper with data. Our goal is, while maintaining the highest level of privilege to internal database admins, we still provide a tamper-evident mechanism.
- **Trust of Third Party:** Some of the commercial solutions provided (Amazon QLDB) assume the organization trusts the third party with protecting its data. Our solution eliminates the need for this trust.
- **No Data Transfer:** Some of the solutions resort to providing an external verifiable copy of the data. This adds some complications like a) confidentiality of data at third party and in transit, b) performance penalty of transferring all data. We eliminate the need for transferring an organization's data and keep it completely in-house.
- **Database:** Some of the solutions protect other objects than databases, for example, documents, filesystem, or even entire operating systems. Our goal is transactional databases.
- **Transactional:** Some of the solutions do protect data but cater more to a batch processing model rather than live transactional systems. We cover the transactional component.
- **Database Specific:** Some of the work provides solutions that have to be implemented in a database specific setup. Even though we have this approach among one of our solutions, we also provide two other alternatives that are completely database agnostic.
- **Transparent:** Some of the solutions are not transparent and require modifications at the application level to function. We provide a solution that is as seamless as possible and that requires zero or very little modifications on the application level. Modifications required at the database level or at the middleware level are minor ones that are add on configurations rather than being invasive. Our goal has been to design a solution that could be transparently retrofitted into existing systems with a) non-invasive approach, and b) empowers old and currently existing systems as well.

Our goal has been to address the abovementioned gaps as much as possible. The reason we have chosen the gaps identified above is that they are vital for any solution to be applicable in existing real industrial use cases rather than just propose a solution that stops only at the theoretical level and falls short of being suitable for solving real life scenarios. Another goal is to provide a solution that does not impractically require total change in an underlying infrastructure.

4 Proposed Solution

4.1 Solution Brief

In our presented solution we build a transparent and seamless middleware for securing database transactions against possible tampering by individuals who have full administrative access to the database and all its related infrastructure. The way this is to be achieved is by leveraging some features of the technology similar to blockchain to interweave sequences of transactions in an unbreakable chain. This is to be done by generating a unique hash for each transaction and using it in a chain of transactions. Any attempts to modify previously entered data will break the hash and therefore the sequence of transactions following such transaction will be invalidated.

In order to guarantee that such a chain could not be regenerated following any tampering attempt, an external source is used for time stamped signing of hashes. The external time-stamp signer is external to the entity so it is beyond the reach of any internal system administrator. Another alternative could be a physical Hardware Security Module (HSM).

In our work, we propose three integration architectures. One is used for Object Relational Mapping frameworks (ORM), the second is for direct database integration, and the third is implementing microservice solutions by a totally transparent reverse proxy.

4.2 The Hasher and The Time-Stamping Signer

The direction adopted is to introduce an externalized time-stamper/signer and/or a tamper-resistant HSM (Hardware Security Module). The role of the signer is to sign a hash of each record/transaction that gets added to the database. In addition to the record, a hash of the previous record is added. A time-stamp is also added to the signed data to protect against future signing replay attacks.

The solution relies on the introduction of a third-party signing authority. The third-party is an external entity that is outside the reach of organization insiders and thus reduces and ideally eliminates the possibility of collusion among internal and external stakeholders.

4.3 Externalized Signer/Stamper

As illustrated in Figure 4, the signer is by design to be external and to serve (as a service) multiple unrelated organization. This adds more security and dramatically reduces the possibility of collusion among system administrators of all the organizations serviced by the signer.

We introduced in Figure 5, an independent signer and time-stamper service (in red). The signer/time-stamper is a totally external entity that could even be outside the organization. The signer service could cater to different organizations as illustrated in the diagram.

In addition to being an external entity, the signer is designed to operate in a completely stateless manner. DBKnot does not rely on the signer keeping any information regarding the data being signed or its corresponding hashes. Such statelessness makes the following possible:

1- **Simplicity of design:** Reduces the range of possible attack vectors making it less vulnerable to attacks.

2- **No Storage – Confidentiality:** No storage is needed on the signer end which adds to security and privacy. This provides zero knowledge securing of the data since it only acts as a signer and not as a repository or secondary storage service.

3- **No Data Transferred:** Actual data never leaves the premises of the user. Alternatively, only a hash is exchanged for the signing process. This reduces a) the network traffic and overhead due to data transfer, b) vulnerability of data in transit to both exposure as well as tampering, and c) having to trust the external signing party on all organization's data.

4- **Workload Balancing:** Statelessness makes it possible to balance loads across many signer nodes as needed if their clocks are well synced. This makes it easy to scale the signing service by adding more servers and distributing the workload among those servers.

5- **Multi-Site Failover:** Statelessness also allows signers to be rolled out at multiple different sites. This provides added reliability in the case of a failure of a whole site due to a total internet outage or a blackout in the hosted area/country.

6- **Proximity:** Statelessness allows servers to be distributed in a way that increase proximity to the users of the servers. This reduces signing latency and duration cost of network delays. This approach is commonly used by Content Distribution Networks (CDNs).

The hasher is the first step of the process. As soon as a record is appended to any of the tracked tables, a hashing process is triggered. The hasher takes the inserted record, creates a structure that represents the concatenation of all fields, hashes that structure, and inserts all information describing that record in the hash table.

Once a record has been added, and after it has gotten automatically hashed, the corresponding hash record will be passed to the signer. The signer will take the hash record, add to it the preceding record together with a time-stamp and sign them all with the signer public key. The signature of the preceding record could be appended to the hashed string instead of the hash, but we see that the hash will be sufficient because it will not be possible to tamper with the hash without breaking the signature. The resulting signature and time-stamp will be

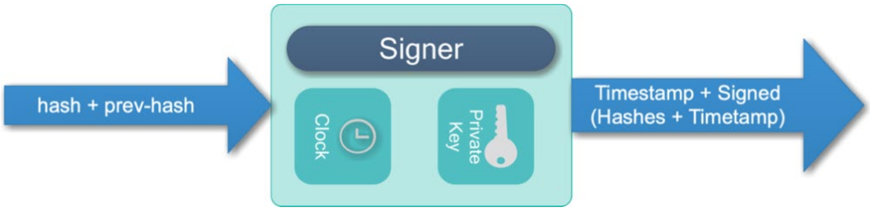


Figure 3: Signer service

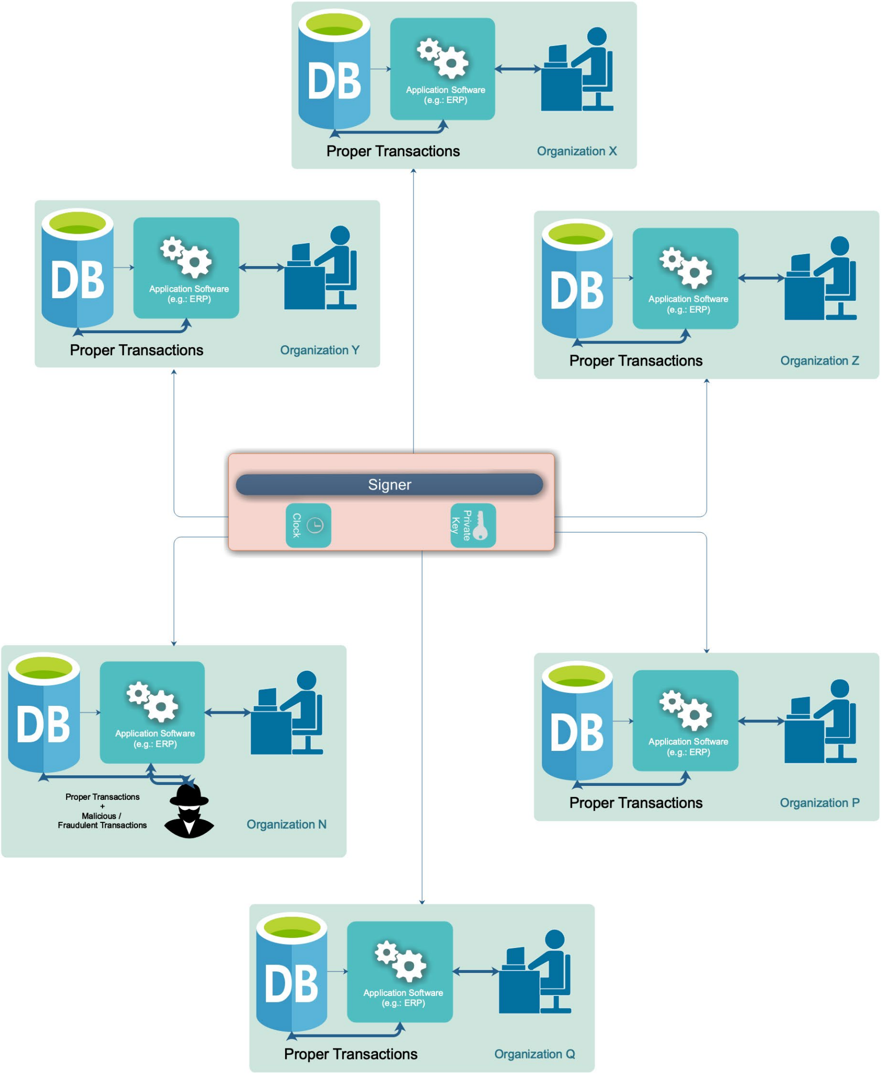


Figure 4: Detailed introduction of a third-party external signing authority

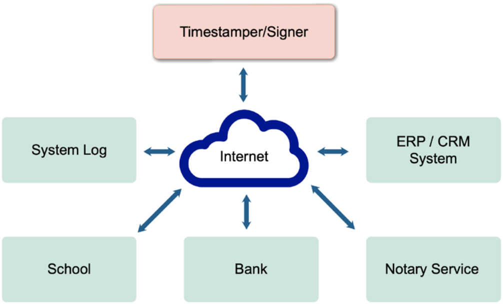


Figure 5: Introduction of a third-party signing service

returned to the database server and stored inside the hash table. The signature saved in the hash table will be used for verification.

4.4 Integration Models

Three different strategies are provided for integrating into existing systems.

The first technique is to design the DBKnot as an embedded layer inside Object Relational Mapping (ORM) systems so application developers can use it seamlessly in a declarative way as detailed below. The second approach is to implement it as a hook into existing databases. This requires less intervention from the user side and only requires an action from the database system administrator. The third and relatively more challenging approach is to be implemented in the form of a REST web service reverse proxy.

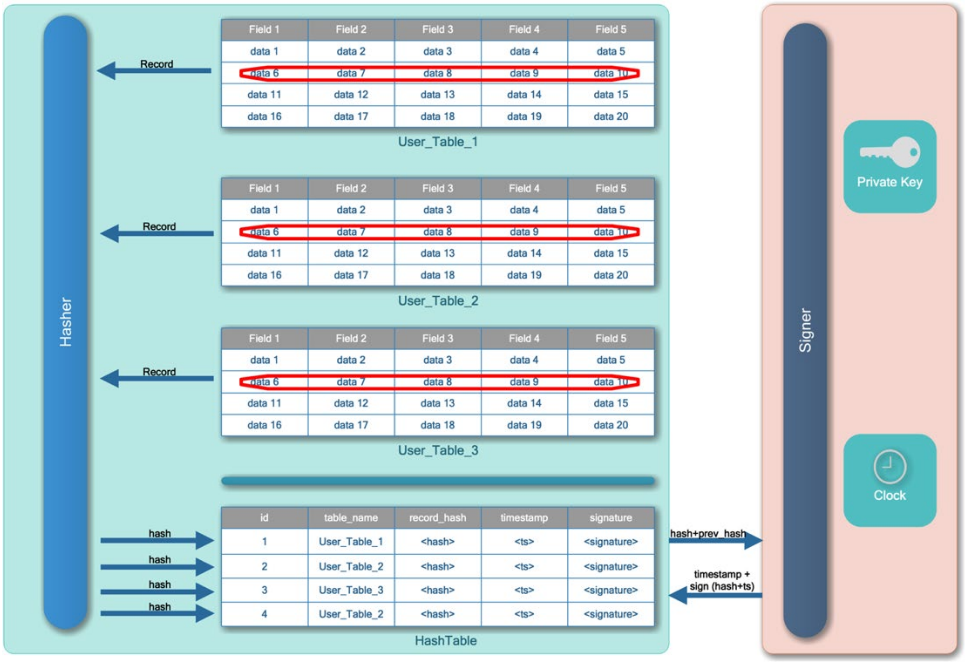
4.4.1 ORM Level Integration. Object Relational Mapping (ORM) frameworks [16-17, 36, 38] sit between developer applications and databases. They provide developers with full object oriented semantics to interfacing with the database.

ORM frameworks allow system developers to use object oriented design to model their data without having to worry about how this maps to the database. ORM frameworks in turn take care of the mapping between data objects on one hand, and tables and relations on the other hand.

At design phase, the ORM layer is responsible for generating the Data Definition Language (DDL) necessary to create the required tables. In SQL these are SQL INSERT statements. The ORM takes care of choosing the necessary dialect of the underlying database by utilizing individual "drivers" for different databases.

ORM layers are also responsible for maintaining the consistency of the mapping throughout the development cycle by propagating any changes done to the model to be reflected immediately into the database structure while preserving all data. This is a process that some implementations call "migration" after the mapping is done, and during runtime, the ORM layer implements all OOP Create, Retrieve, Update, and Delete (CRUD) operations by mapping them to their corresponding Data Manipulation Language (DML) statements. In SQL, this is done by using INSERT, SELECT, UPDATE, and DELETE SQL statements respectively. As done in the DDL, all DML statements are generated by the ORM driver that corresponds to the database being used which in turn ensures that the necessary SQL flavor is used.

In addition to the declarative semantics and ease of use by developers, embedding a tamper-detection layer inside the



Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

User_Table_1

Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

User_Table_2

Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

User_Table_3

id	table_name	record_hash	timestamp	signature
1	User_Table_1	<hash>	<ts>	<signature>
2	User_Table_2	<hash>	<ts>	<signature>
3	User_Table_3	<hash>	<ts>	<signature>
4	User_Table_2	<hash>	<ts>	<signature>

HashTable

Figure 6: Signer and time-stamper

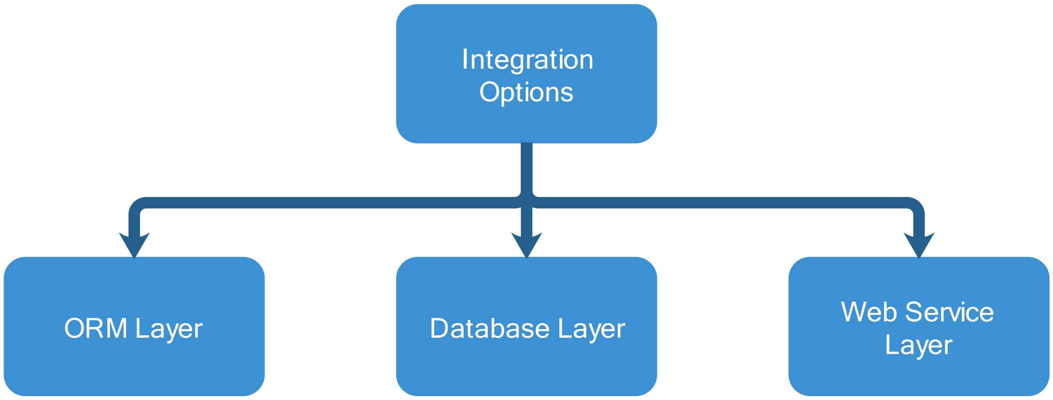


Figure 7: Integration options

ORM layer also makes it completely database agnostic.

Meaning that the same implementation will work on any database as long as it is supported by the used ORM layer without any changes.

Figure 8 shows how the ORM layer sits between the developer code and the database itself and abstracts away all of the DBMS specific relational database operations.

In the following section, two different techniques are outlined for integration into ORM systems.

The first one is through an application level ORM interceptor, and the second one is through implementing a framework level global middleware to perform the ORM functionality.

4.4.1.1 ORM Technique 1: ORM Interceptor. To retrofit DBKnot functionality into an ORM application, as the user code initiates any persistent database operations (insert operations) that are tagged as trackable, the ORM interceptor takes the transaction, passes it to the original ORM layer which takes care of the transaction as normally expected. Afterwards, the ORM interceptor starts doing its own hashing and signing actions by hashing the record and adding it into a local hash table and then communicating with an external signer to sign the transaction and save the signed hash linked with the previous hash.

Figure 10 shows how the DBKnot hook is inserted in the middle of the operation. DBKnot intercepts all calls to the ORM, performs the needed hashing and signing functionality, and passes execution to the original ORM framework.

The integration layer is designed to provide a completely seamless user experience to developers. In the current implementation, as illustrated in Figure 11, all a user (developer) needs to do is to have his/her model classes extend a class (a mixin) that provides all needed functionality.

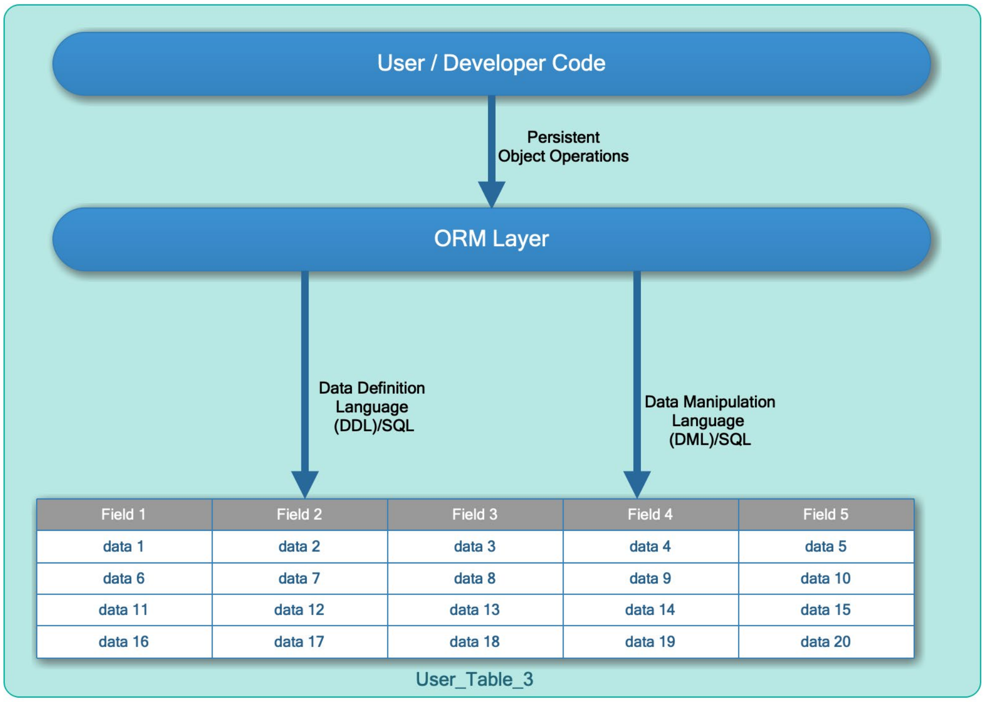
4.4.1.2 ORM Technique 2: Framework-wide Global Middleware. A second approach to integrating into ORM systems is to integrate in the form of a middleware that is embedded into the ORM framework itself. The advantage of this approach is that it is completely transparent and will not even require the declarative approach of extending a "trackable" class in system code. The side effect however of this approach is that it will give application developers less control to selectively track specific models (tables) while ignoring the tracking of other models. This could be mitigated through the implementation of an external configurator that could be managed separately to disable universal tracking and allow selective tracking of data models.

4.4.1.3 More Efficient ORM Tracking through Parallel Pipelining. The efficiency of the previously outlined ORM tracking could be increased through the introduction of a level of parallelism. Such parallelism in signing and stamping is not going to be as simple as just creating a parallel thread due to the fact that the feature of "chaining" introduces dependencies. Due to this level of dependency, a pipelining technique is introduced as detailed in Chapter 4.7.

Figure 12 shows an adapted version of the activity diagram after adding the parallel tracking.

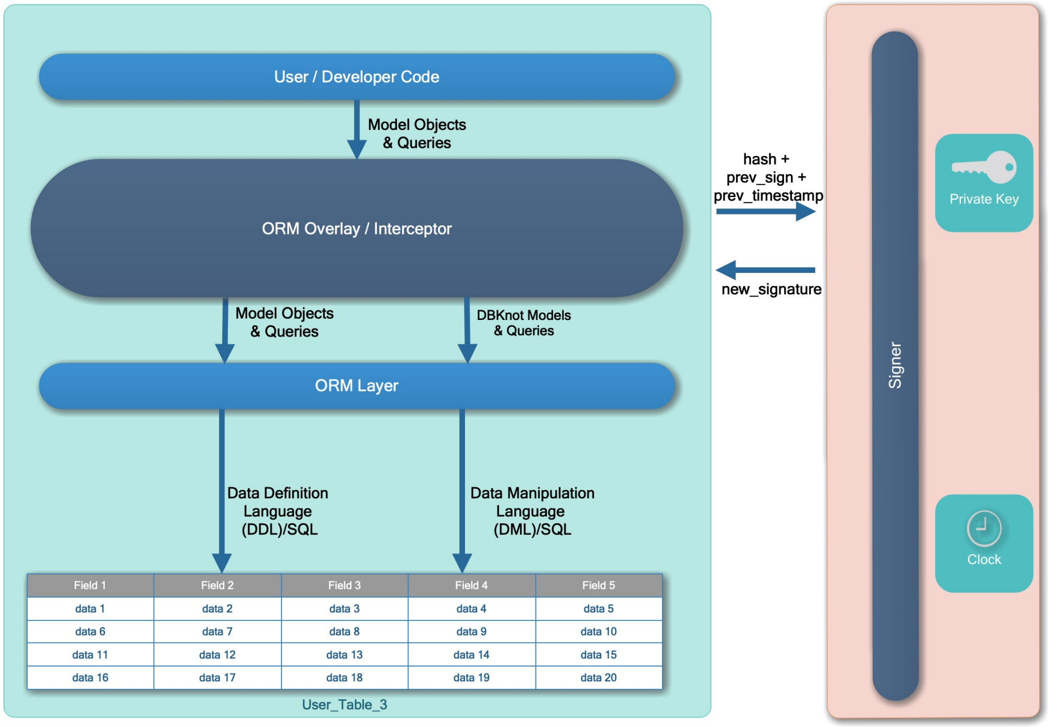
4.4.2 Database Level Integration

DBKnot also supports database level integration. This is done



Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

Figure 8: Standard ORM operations



Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

Figure 9: ORM interceptor

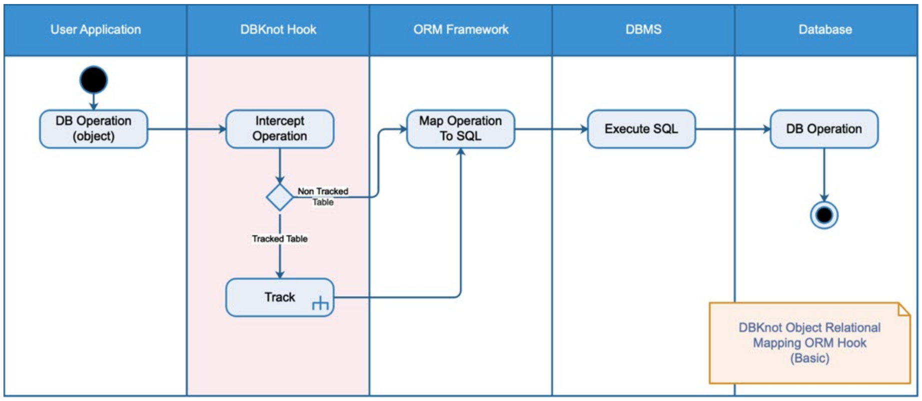


Figure 10: Adding DBKnot ORM hook basic activity diagram

```
class Test(DBKnotMixin):
    name=models.CharField("Name",max_length=50)
    def __str__(self):
        return self.name
```

Figure 11: ORM simple mixing implementation

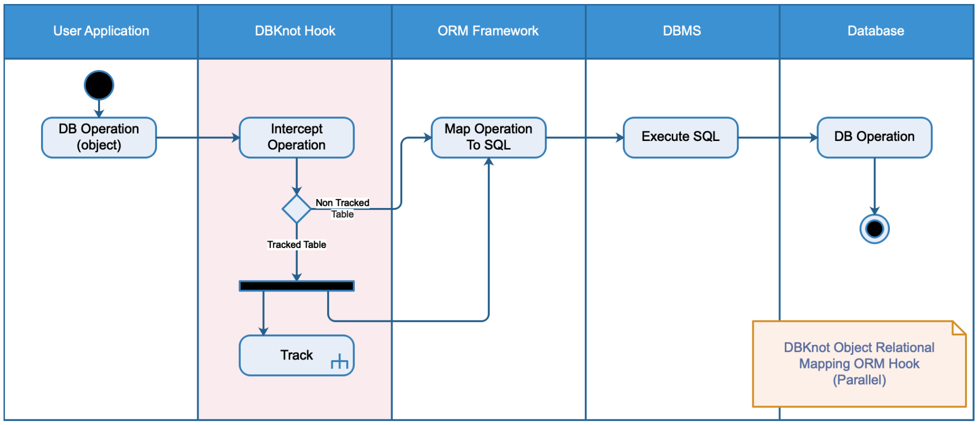


Figure 12: Adding DBKnot ORM hook parallel activity diagram

by embedding triggers on tracked tables. When a record is inserted in a tracked table, the trigger will be fired and will perform all the needed tracking functionality.

The default behavior in Figure 13 is changed by adding the DBKnot layer. The DBKnot layer is called a database trigger that tracks desired tables.

Figure 14 shows the asynchronous version of the DBKnot database level integration where the hashing and signing functionality is signaled by a trigger in the database level.

4.4.2.1 The Signer. The direction adopted is to introduce an externalized time-stamper/signer and/or a tamper-resistant HSM (Hardware Security Module). The role of the signer is to sign a hash of each record/transaction that gets added to the database. In addition to the record, a hash of the previous record will be added. A time-stamp is also added to the signed data in order to protect against future signing replay attacks.

4.4.2.2 A Chain of Hashes. A chain of the hashed transactions is being maintained. The chain includes the signed hashes of the data as well as the time-stamps. Each record will include a hash of the previous record.

The chain of hashes is the only item that is added to the existing database. All other tables, field definitions, and records are untouched and remain intact.

As illustrated in Figure 16, The hash-chain-table is made up of the following fields:

1- **id:** A sequential ID. This is very important for identifying the sequence of transactions hashed. This is used during the signing and signature verification process.

2- **table_name:** The name of the table where the record came from. The hashing table is a database wide table. Meaning that it contains hashes of all records regardless of which table they come from. This keeps the hashing table as the only item added to the database and avoids making any changes of any other tables of the database to be secured.

3- **record_hash:** A hash of the record chained is placed in this field. In this research, MD5 hashing has been used. It is necessary that a fast hashing algorithm is used. Hashing is

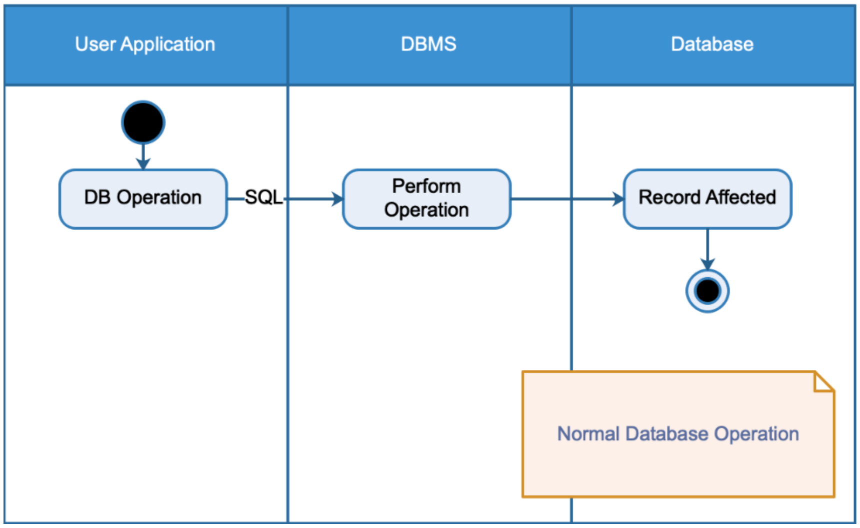


Figure 13: Normal database operation

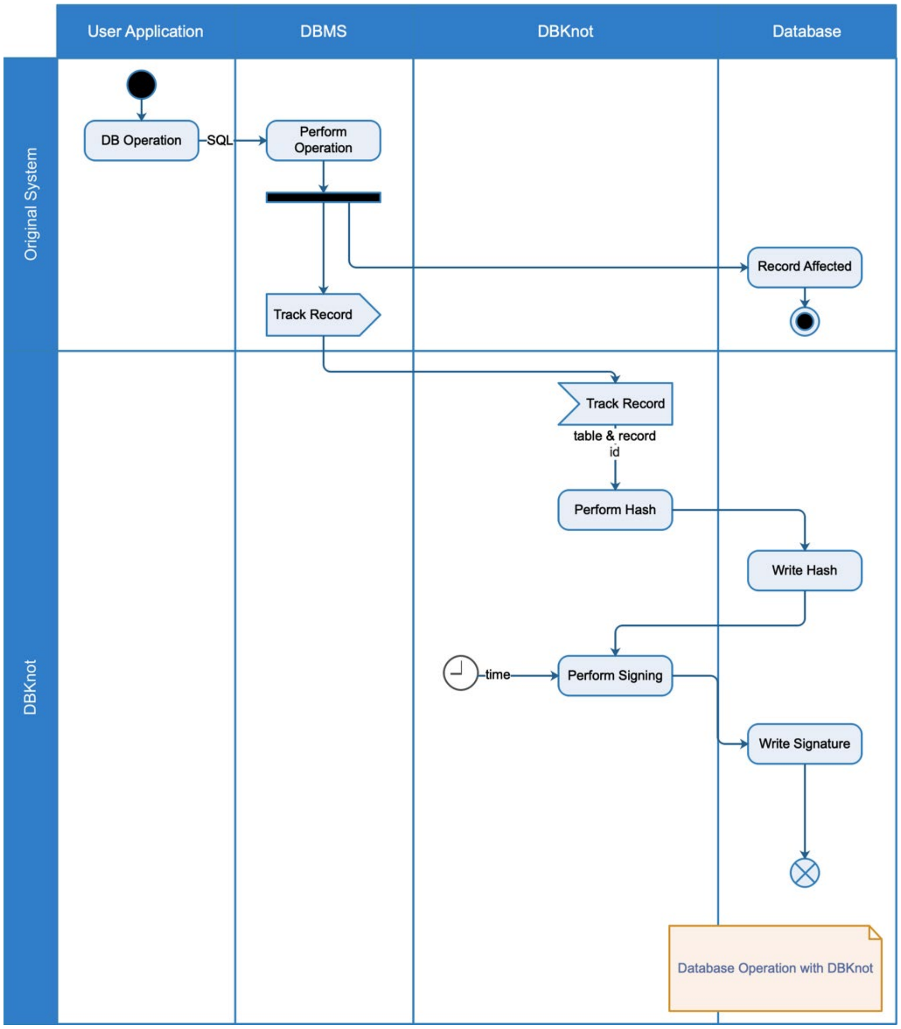


Figure 14: Database level DBKnot integration

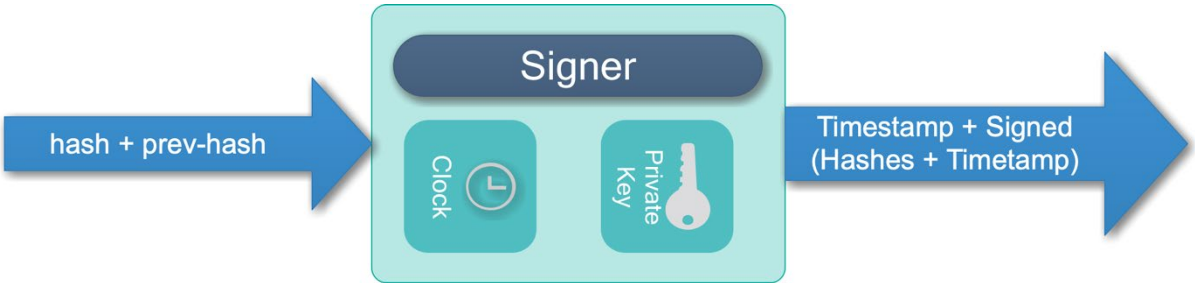


Figure 15: Signer service

applied to a structure that contains a concatenated form of all record fields. SHA-256 or 512 could also replace MD5 for added security but with their corresponding performance tradeoff [9, p 2]. We believe however, that such a change may or may not be necessary depending on the application. It is not practical (in fact almost not possible) to generate a reverse hash for a specific number or piece of information that needs to be tampered. The only possibility here will be to generate a reverse hash to corrupt the data rather than put in any meaningful data. Again, it could be configurable and left to be decided on a case-by-case basis.

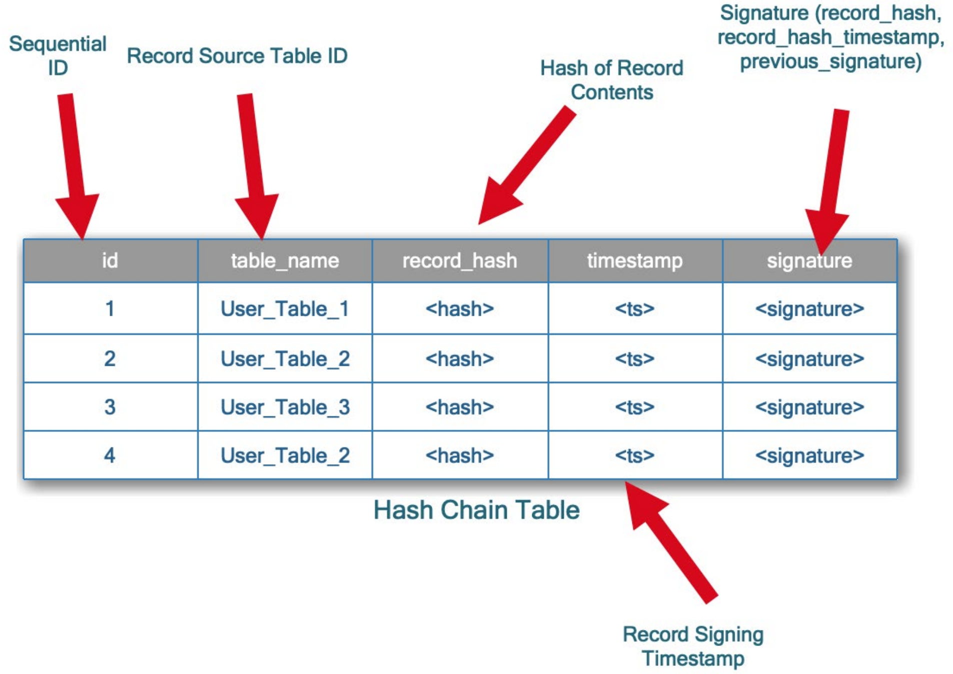
4- **Time-stamp:** This field is filled by the data returned from the signer. It is the signature time-stamp.

5- **signature:** In this field, the signature itself is stored as returned by the signer.

4.4.2.3 The Hasher. The hasher is the first step of the process. As soon as a record is appended to any of the tracked tables, a hashing process is triggered. The hasher takes the inserted record, creates a structure that represents the concatenation of all fields, hashes that structure, and inserts all information describing that record in the hash table as described in Section 4.4.2.2.

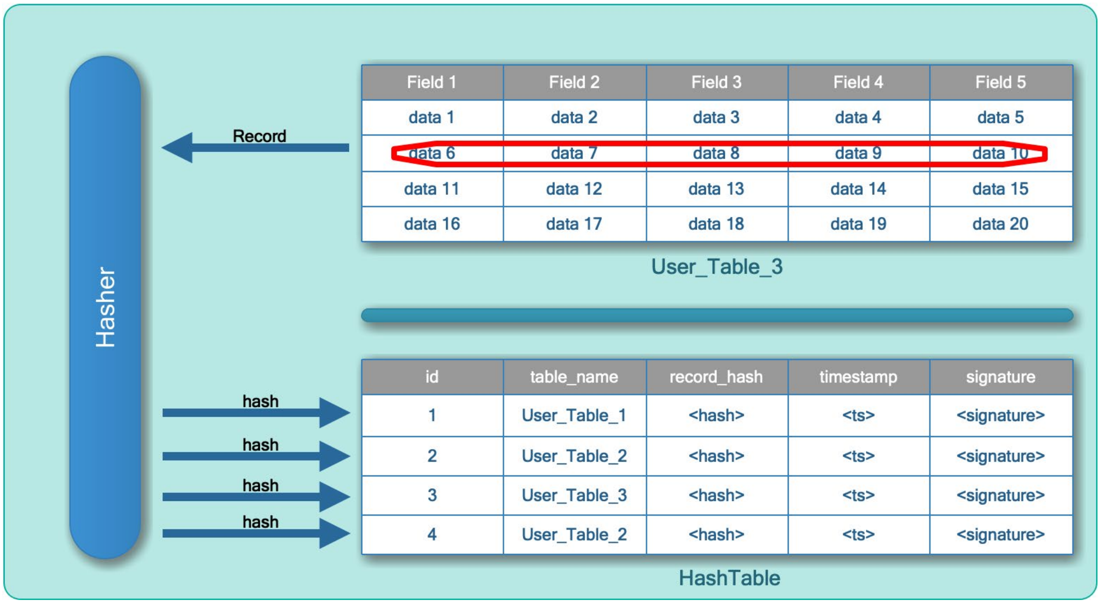
Parallelizable Hashing: By nature, the hashing process is parallelizable. This will utilize any available parallelism infrastructure present at the database server to optimize signing performance. In addition, it could be done by any external server that has access to the same database or a live replica of the database to relieve the primary server from extra computation work.

4.4.2.4 Inserting the Signer and Time-Stamper. Once a record has been added, and after it has gotten automatically hashed, the corresponding hash record will be passed to the



id	table_name	record_hash	timestamp	signature
1	User_Table_1	<hash>	<ts>	<signature>
2	User_Table_2	<hash>	<ts>	<signature>
3	User_Table_3	<hash>	<ts>	<signature>
4	User_Table_2	<hash>	<ts>	<signature>

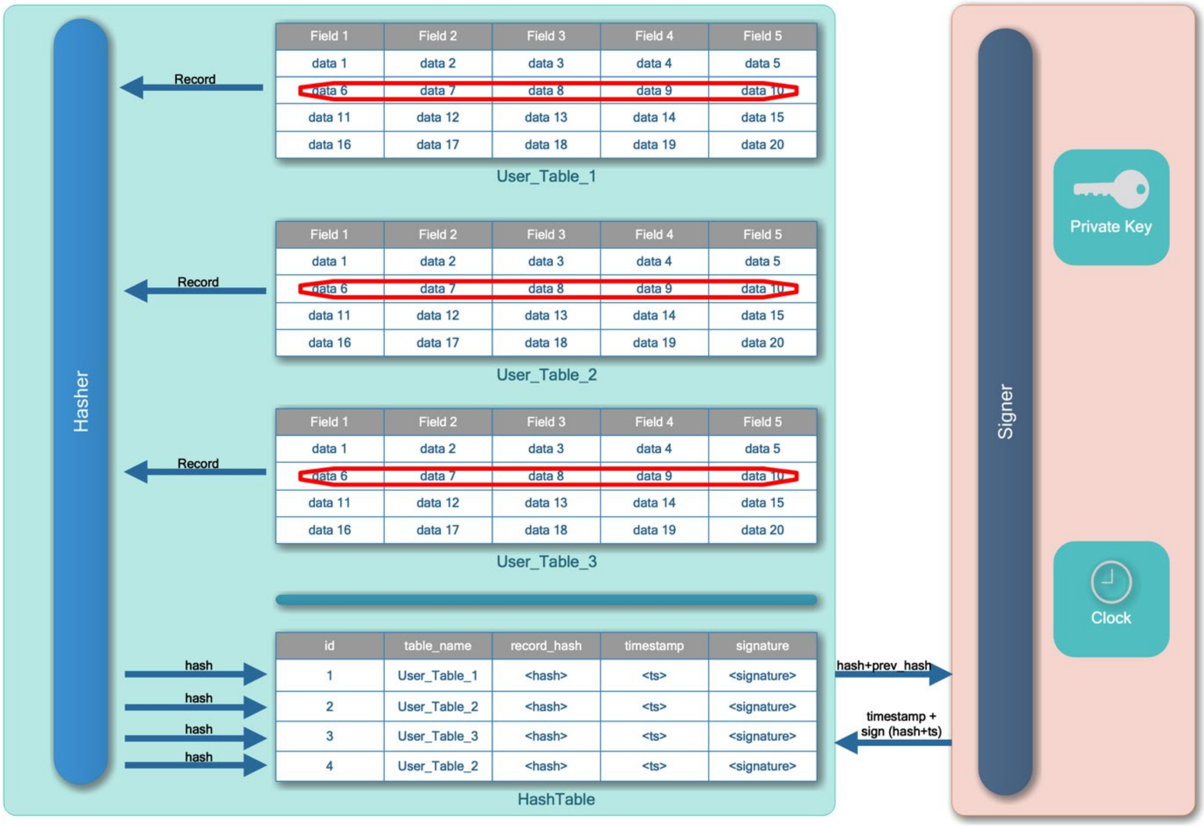
Figure 16: Chain of hashes



Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

id	table_name	record_hash	timestamp	signature
1	User_Table_1	<hash>	<ts>	<signature>
2	User_Table_2	<hash>	<ts>	<signature>
3	User_Table_3	<hash>	<ts>	<signature>
4	User_Table_2	<hash>	<ts>	<signature>

Figure 17: Hasher



User_Table_1

Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

User_Table_2

Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

User_Table_3

Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

HashTable

id	table_name	record_hash	timestamp	signature
1	User_Table_1	<hash>	<ts>	<signature>
2	User_Table_2	<hash>	<ts>	<signature>
3	User_Table_3	<hash>	<ts>	<signature>
4	User_Table_2	<hash>	<ts>	<signature>

Figure 18: Signer and time-stamper

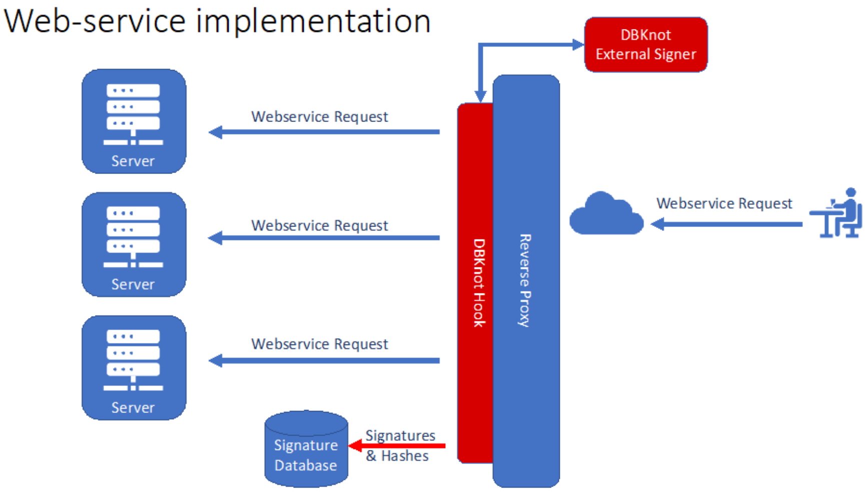


Figure 19: Web service implementation

signer. The signer will take the hash record, add to it the preceding record together with a time-stamp and sign them all with the signer public key. The signature of the preceding record could be appended to the hashed string instead of the hash, but we see that the hash will be sufficient because it will not be possible to tamper with the hash without breaking the signature. The resulting signature and time-stamp will be returned to the database server and stored inside the hash table.

The signature saved in the hash table will be used for verification.

4.4.3 Web-Service/API Microservices Architecture. DBKnot functionality could be implemented inside a middleware. The benefit of injecting the functionality in the form of a middleware is that it could allow the functionality to be retrofitted into existing applications while doing zero changes to the existing application. This way existing applications can benefit from DBKnot and secure their data seamlessly.

This approach is better suited to cater to applications with microservice architectures.

Challenge: This will require an easy-to-use mini language/syntax for application developers to define their application web service's semantics.

Advantage: Totally non-invasive, could be totally external to server inside a reverse proxy.

In this approach, the DBKnot functionality is to be implemented in the form of a reverse proxy/middleware that sits between all incoming API requests and the system being tracked.

The following are the advantages of implementing DBKnot in the form of a web service intermediary:

- **Technology agnostic:** Totally decoupled from any underlying technology used by the software implementation.
- **Supports hybrid microservices:** In an enterprise application or a set of applications that is dependent on numerous microservices, this design will be able to support all of the services even if they are implemented by different software/applications (e.g., billing software + accounting software + CRM software, etc.)
- **Multi-server support:** This approach will function regardless of the number of back-end servers providing the service. It will also work in load balancing use cases.
- **Non-relational Database:** Relying on REST web services for tracking database CRUD operations opens the way to cater to other non-relational database models directly without being limited to a particular ORM framework or a database management system.

The drawback/challenge however to implementing DBKnot as a web service is the lack of adherence to a concrete and clear CRUD standard in the usage of REST web services. Accordingly, such implementation will need to be configurable to match each service that it intercepts. So, even though the original software is untouched, work will need to be done at the reverse proxy level in order to configure DBKnot, and this will make it implementation specific.

As mentioned in Section 2.2, our approach is to try and base record chaining on the semantics of using the REST API to do CRUD functionality. This is a good entry point to the implementation of this technique. The technique could be taken a step further into covering other REST semantics but will require more implementation specific configuration and will be less transparent.

The following are some REST methods that are based on standard HTTP methods: [21, 32].

As we see in Table 1, HTTP (REST) methods automatically lend themselves to data operations.

Additionally, most of the HTTP (REST) response codes match standard database operations. [32]

4.4.4 REST API Based Definition. To be able to track a microservice based request, in most cases a specific configuration is required. Fortunately, there are new industry standards [3] for performing such configurations. Examples are OpenAPI [2, 19, 33] and RAML [25].

As we can see, a number of the details of the possible web service operation is specified in YAML format.

4.5 Verification Steps

Verification of records and thus, the detection of possible tampering falls into the following three categories:

Table 1: HTTP methods and REST

Method	Use
GET	Retrieve a particular record of data
HEAD	Get a summary of record data
PUT	Add a record
POST	Possibly update a data record
DELETE	Delete a data record

Table 2: HTTP (and REST) return codes

HTTP Return Code	Meaning
200 OK	Operation performed correctly
201 Created	Record added correctly
400 Bad Request	There is a problem with the request
401 Unauthorized	Authentication Required
403 Forbidden	User permission problem
404 Not Found	Item being queried does not exist

This is an example service definition using OpenAPI:

```
tags:
- pet
summary: Updates a pet in the store with form data
operationId: updatePetWithForm
parameters:
- name: petId
  in: path
  description: ID of pet that needs to be updated
  required: true
  schema:
    type: string
requestBody:
  content:
    'application/x-www-form-urlencoded':
      schema:
       type: object
       properties:
          name:
            description: Updated name of the pet
            type: string
          status:
            description: Updated status of the pet
            type: string
       required:
         - status
responses:
  '200':
    description: Pet updated.
    content:
      'application/json': {}
      'application/xml': {}
  '405':
    description: Method Not Allowed
    content:
      'application/json': {}
      'application/xml': {}
security:
- petstore_auth:
  - write:pets
  - read:pets
```

1- Malicious addition of a record: results in a record that does not have a corresponding signed hash in the hash/signature table.
2- Malicious deletion of existing records: results in an existing hash/signature without a corresponding record.
3- Malicious tampering with hashes or signatures: results in a scenario that is a combination of the two tampering situations above.

Figure 21 shows an example of the inconsistencies resulting from maliciously adding a record to the database.

There are two cases when a verification is triggered. The first one is at data read or insertion time where one record needs to be verified. The verification step will trace the record back throughout the chain through an “n” predefined depth before generating the assumption that it was not tampered with within a particular time window (1 week, 1 month, 1 year, etc.).

The second case is the case of patrolling threads/processes. These are housekeeping threads that regularly patrol the database to check and confirm the correctness of all records, hashes, signatures, and linkages.

We believe more work could be done on both verification cases to optimize such a process and increase the coverage of tests within the same short duration of time.

4.6 Performance Optimization

The additional tracking/hashing/signing layer does not come without an expense. There is of course a performance impact on insert transactions into the database. In this section we illustrate a number of different optimizations that could be used to mitigate and reduce such an impact. Most of them will be for the purpose of introducing different forms of parallelism into the design.

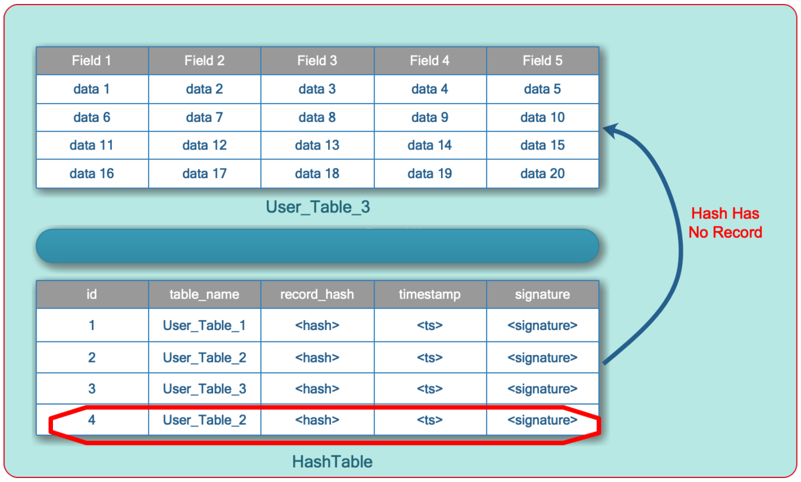
4.6.1 Signing Distribution. In this design illustrated in Figure 22: Parallel signers - consistent hashing, a technique similar to database record sharding is used to distribute workload on a number of different shards. Instead of chaining signed blocks in a purely sequential manner, they are chained in a round robin form. In this case, if the system is configured to use **“n”** shards, then each record **“i”** will be chained with distributed to shard **“$s = i \% n$”**. The record will be linked to the previous record in the same shard too. Please note that the “I” is the sequence ID of the hash record rather than the ID of any of the tables. So, there is no possibility of collisions with other IDs in the system.

The advantage of this technique is that it breaks down the added latency and sequentiality of the process and introduces a degree of parallelism. Utilizing this method, several insert statements together with their corresponding hashes could be done in parallel without having to wait for each other to finish.

The tradeoff in this approach is that database verification is divided into “n” independent chunks which makes the chaining process less complex. One mitigation for that is to introduce occasional inter-shard linkages to tightly intertwine them together and eliminate that independence.
Figure 23 illustrates how consecutive transactions are linke, hashed, chained, and signed together and how they are split into groups.

4.6.2 Coarse Grained Block Signing. Instead of performing hashing and signing on a record-by-record level, records are grouped into blocks. Each block is hashed together and then the group hash is signed by the signer.

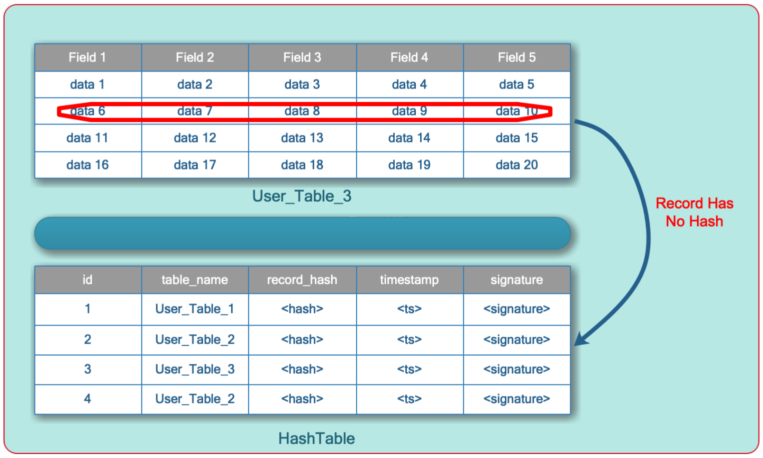
The figure below (Figure 24) shows how transaction batches are broken down into blocks and each block is hashed and signed separately. This approach reduces the signing overhead and enhances performance. Instead of a hash table with an entry for every record, a smaller hash table is utilized with a record per batch. There is a tradeoff however between the batch (block) size and the time required to verify a record.
Another drawback is that records of a whole batch will remain untracked until the batch is completed and signed. This will be problematic in cases where the database undergoes few transactions. To mitigate for this problem, a variable size block could be implemented (illustrated in Figure 24: Coarse grained signing - variable block size) where if a block remains open for a certain (configurable) duration of time, the system generates a clock event. This clock event with its corresponding time-stamp will force the closing and signing of the open block regardless of the number of records in the block. This approach will also have the added benefit of being able to work in an environment



Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

id	table_name	record_hash	timestamp	signature
1	User_Table_1	<hash>	<ts>	<signature>
2	User_Table_2	<hash>	<ts>	<signature>
3	User_Table_3	<hash>	<ts>	<signature>
4	User_Table_2	<hash>	<ts>	<signature>

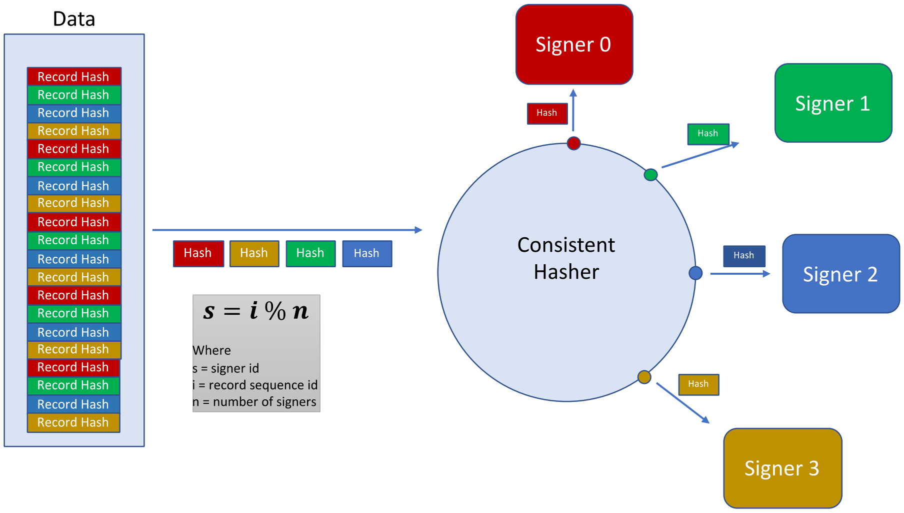
Figure 20: Detection of a maliciously deleted record



Field 1	Field 2	Field 3	Field 4	Field 5
data 1	data 2	data 3	data 4	data 5
data 6	data 7	data 8	data 9	data 10
data 11	data 12	data 13	data 14	data 15
data 16	data 17	data 18	data 19	data 20

id	table_name	record_hash	timestamp	signature
1	User_Table_1	<hash>	<ts>	<signature>
2	User_Table_2	<hash>	<ts>	<signature>
3	User_Table_3	<hash>	<ts>	<signature>
4	User_Table_2	<hash>	<ts>	<signature>

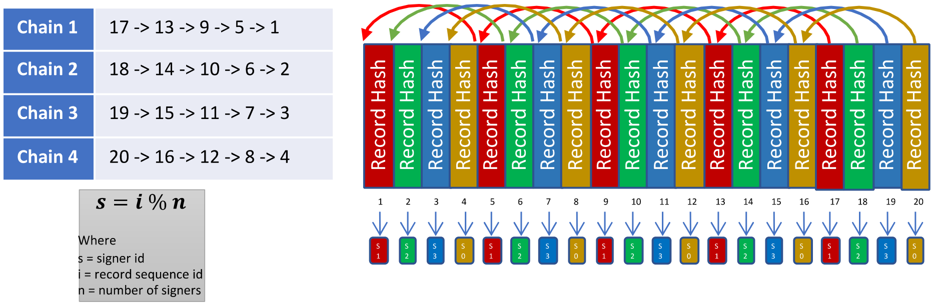
Figure 21: Detection of a maliciously added record



$s = i \% n$

Where
s = signer id
i = record sequence id
n = number of signers

Figure 22: Parallel signers - consistent hashing



Chain 1	17 -> 13 -> 9 -> 5 -> 1
Chain 2	18 -> 14 -> 10 -> 6 -> 2
Chain 3	19 -> 15 -> 11 -> 7 -> 3
Chain 4	20 -> 16 -> 12 -> 8 -> 4

$s = i \, \% \, n$

Where
s = signer id
i = record sequence id
n = number of signers

Figure 23: Parallel signers - linking of hashes

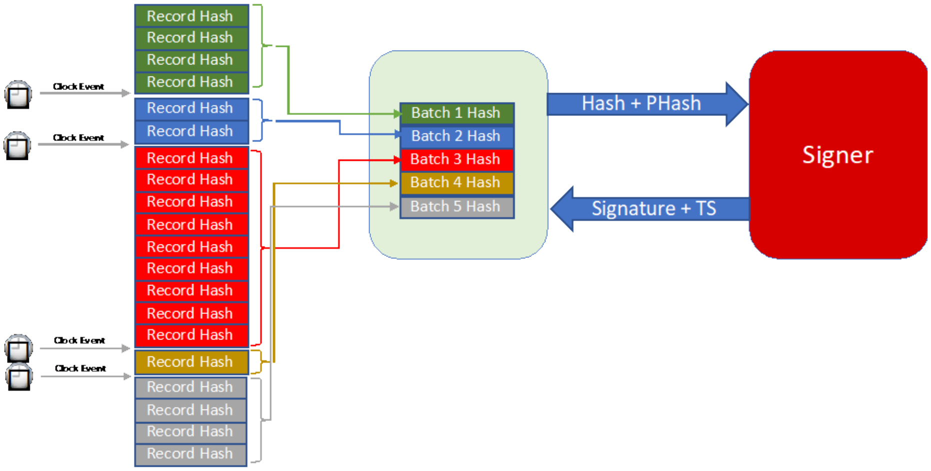


Figure 24: Coarse grained signing - variable block size

with intermittent or unreliable connectivity.

4.7 Performance Optimization – Pipelining

Four different techniques are being used for handling sequentiality/parallelism in implementing the DBKnot chaining process.

The first technique is purely sequential, the second technique pipelines the signing process, the third technique pipelines both the hashing and signing processes combined, and the fourth technique designs everything to be pipelined.

Each one of the techniques will be further explained in its own corresponding section.

4.7.1 Parameters. For each of the techniques used, there are three assumed scenarios that will be tested. All the scenarios are variants of the following set of variables:

- **Transaction time:** The time taken to perform a transaction on the database.
- **Hashing time:** The time taken to hash a transaction.
- **Signature time:** The time taken to sign the hashes and produce a signature.

All variables

$$
\begin{aligned}
&n = number\ of\ transactions\\
&t = transaction\ time\ (t1\\
&\qquad\qquad \rightarrow short\ transaction, t2\\
&\qquad\qquad \rightarrow long\ [4X]\ transaction)\\
&h = hashing\ time\\
&s = signing\ time\\
&v = total\ batch\ duration
\end{aligned}
$$

Figure 25: Testing variables

The following categories of transactions were derived from the preceding variables:

- **Transaction Bound:** In these scenarios, the transaction time is the longest of the three numbers.
- **Hashing Bound:** In these scenarios, the hashing time is the longest of the three numbers.
- **Signing Bound:** In these scenarios, the signing time is the longest of the three numbers.

All tests are done on two batches of transactions, one of them is made up of transactions that require a small “t1” to run, another one is a long batch with transactions taking longer time “t2” where ($t2 = 4 \times t1$). There are two other intermediate batches but we have decided to not include their results in this document due to the sufficient clarity of the other samples.

4.7.2 Technique 1: Inline Hashing & Signing. The first technique used is to perform the transaction, followed by the hashing process, followed by the signing process. They are all done in series as illustrated in Figure 26.

There are three scenarios of implementing the “all-inline” sequential method. Such scenarios are used in comparison of different techniques under varying conditions.

The formula in Figure 27 shows that due to the linear dependency nature of this approach, the total time taken is a simple sum of the total time taken for each transaction (transaction time “t” + hashing time “h” plus signing time “s”) and that the process is a very basic sequential one without any performance gains from any potential parallelism.

4.7.3 Technique 2: Partial Concurrency Through Signature Pipelining. This technique removes the signing process out of the main execution pipeline to allow running it in parallel when needed to gain some performance. Please note that the transaction and hashing in this approach remain sequential.

4.7.4 Technique 3: Concurrency Through Hash and Signature Pipelining. This technique separates the hashing and signing from the main thread and executes them separately in a single thread of sequential execution. Please note that they are both sequential as well. The signing process has been increased in duration to illustrate the sequential nature of the process and its impact.

4.7.5 Technique 4: Concurrency Through Pipelining All Operations. This technique is different from all the others above. In this technique we separate each of the three steps (transaction, hashing, and pipelining) into its own pipeline and let them run asynchronously while preserving sequence dependencies.

In this solution everything runs in parallel. Where a hasher is separate from a signer and separate from the main transaction thread of execution.

5 Experimentation and Results

Workloads were automatically generated by taking into consideration covering all different combinations of different inputs. For example, signing time was generated to include a whole spectrum of signing time displaying the existence of local vs. remote signer and different delays in the signing process. The same was done for the hashing time as well as transaction time.

The two comparison sets of heatmaps below show that pipelining does enhance performance in most cases. The following is a summary of the pipelining results:

All inline

- Base performance.
- Increase in record hashing or signing time results in equal impact on performance.

- Pipeline signing
  - Better overall performance
  - Increase in signing time results in less performance degradation than increase in hashing time due to parallelism.

- Pipeline signing & hashing
  - Slight performance improvement from the signing-only pipelining.
  - Equal impact of increase in hashing and signing time on the total duration.

- Pipeline all
  - Significantly better performance.
  - Performance is slightly better when hashing and signing time are similar.

6 Conclusions and Future Work

As a conclusion, and after going through related work in the same area, we believe we have added a new solution for tamper detection for a certain class of problems. The solution is designed to be very lightweight, easy to retrofit into existing systems, as well as adding almost zero steps requiring handling data either in transit or in new storages.

We designed a tamper-evident architecture called DBKnot that detects database tampering in most cases. An external signer is being used to further protect the database from tampering even by an insider who has full authority and access rights over the whole system, including operating systems, databases, network, and firewalls. DBKnot enables tracking of individual tables that are immutable such as accounting systems, banking systems, and system logs. A chain of records inspired

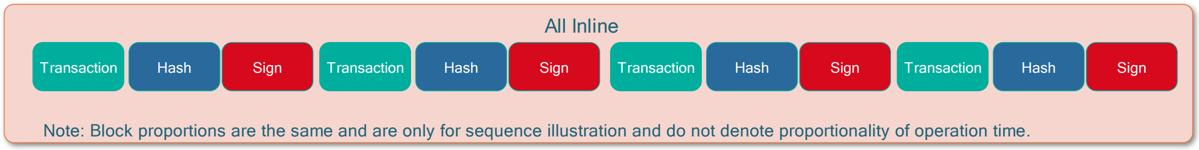

All Inline
Transaction
Hash
Sign
Transaction
Hash
Sign
Transaction
Hash
Sign
Transaction
Hash
Sign
Note: Block proportions are the same and are only for sequence illustration and do not denote proportionality of operation time.

Figure 26: Inline hashing & signing

All inline formula:

$$v = \sum_{i=0}^{n} t + h + s$$

Figure 27: Formula for "all inline"

by blockchain is used to interlink records together through linking their hashes. Each hash link is signed using an external signer or a hardware security module.

We showed how the techniques could apply in three different modes of integration: 1) Embed inside a database management system, 2) Embed inside an Object Relational Mapping framework, or 3) Implement as an external reverse-proxy for multiple web-services and even multiple totally different servers.

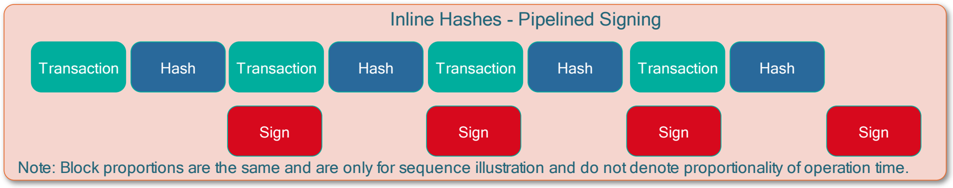
We have illustrated how DBKnot could be implemented in a web service model and how new web service definition languages can be used to facilitate the DBKnot web service configuration process for systems that adhere to the standard and properly define their services. In that case, this can be done with much less intervention from the system admin than if nothing was defined at all.

We have performed tests using generated workloads. As expected, the tests showed an increased overhead for the hashing and signing operations. The overhead though was almost constant when prorated to a transaction level, meaning that it would scale up with the same level of performance. Performance overhead could be significantly reduced by using different parallelization and pipelining techniques to reduce the synchronicity of hashing and signing.

We have explored different parallelization by testing four techniques of parallelization. The first approach was zero parallelization where everything is run in series, and then incrementally started parallelizing step by step until we reached an all parallel scenario. The testing showed that parallelization will lead to a significant performance leap.

The following are some areas that could be enhanced or features that could be added in upcoming related work:

The current work assumes that data being tracked is immutable. Further work can be done by finding different techniques or approaches that would enable catering to database systems that change through updates and deletes with reasonable optimality while utilizing the same technique of relying on external signers for security against internal tampering.


Inline Hashes - Pipelined Signing

Note: Block proportions are the same and are only for sequence illustration and do not denote proportionality of operation time.

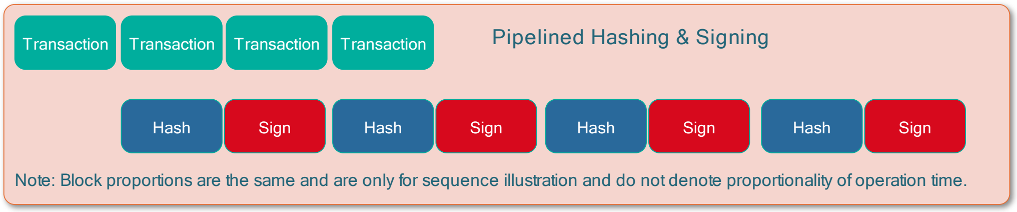
Figure 28: Partial concurrency through signature pipelining

Formula for signature pipelining:

$$v1 = s + \sum_{i=0}^{n} t + h \qquad v2 = t + \sum_{i=0}^{n} s + h$$

$$v = \max{(v1, v2)}$$

Figure 29: Formula for signature pipelining



Pipelined Hashing & Signing

Note: Block proportions are the same and are only for sequence illustration and do not denote proportionality of operation time.

Figure 30: Concurrency through hash and signature pipelining

Formula for hash and signature pipelining:

$v1 = s + \sum_{i=0}^{n} t + h$	$v2 = t + \sum_{i=0}^{n} s + h$
$v = \max(v1, v2)$	

Figure 31: Formula for signature and hash pipelining

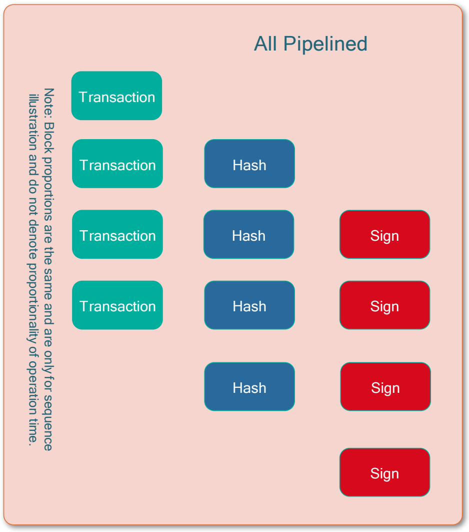

All Pipelined
Note: Block proportions are the same and are only for sequence illustration and do not denote proportionality of operation time.
Transaction
Transaction
Transaction
Transaction
Hash
Hash
Hash
Hash
Sign
Sign
Sign
Sign

Figure 32: Pipelining all operations

The area of Merkel Trees could be studied further. Verification algorithms utilizing a Merkel Tree like approach could result in more efficient verification of tracked records.

More studies need to be done to see how the system can be adapted to changes in database structure. This would enable, not only established and mature systems in production, but also dynamic and changeable systems that are undergoing constant development.

DBKnot is designed as much as possible to detect any tampering with data inside the database. There are however two cases that are not covered. The first case is where the fraudster has access to the application source code. In this case the data is tampered in transit before reaching the database. So the database has no knowledge that the application data has been tampered with. The second vulnerability is the small window between the transaction and the hashing of the transaction. This window could be controlled (shortened or extended) by changing the signing granularity or eliminating block signing altogether and enabling per transaction signing. It is a tradeoff between window size and performance

.

Formula all pipelining:

$v1 = h + s + \sum_{i=0}^{n} t$	$v2 = t + s + \sum_{i=0}^{n} h$	$v3 = t + h + \sum_{i=0}^{n} s$
$v = \max(v1, v2, v3)$		

Figure 33: Formula for pipelining all operations

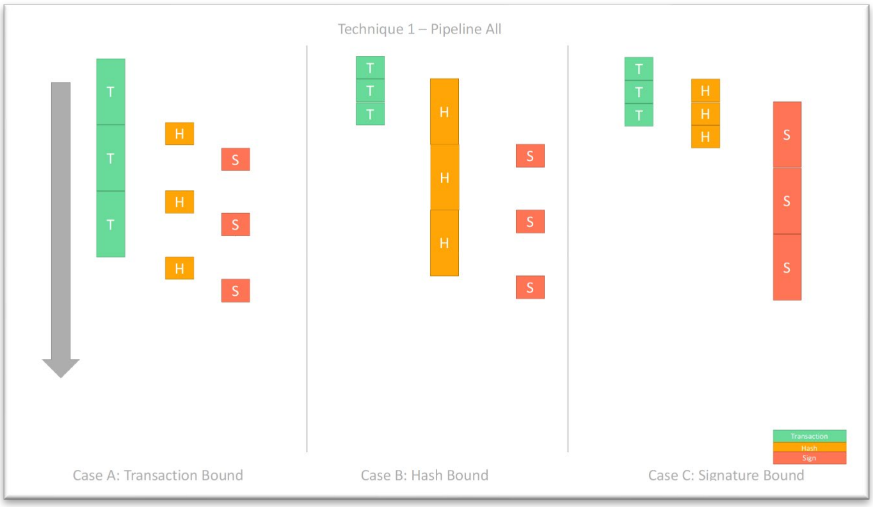


Figure 34: Pipelining all – illustration

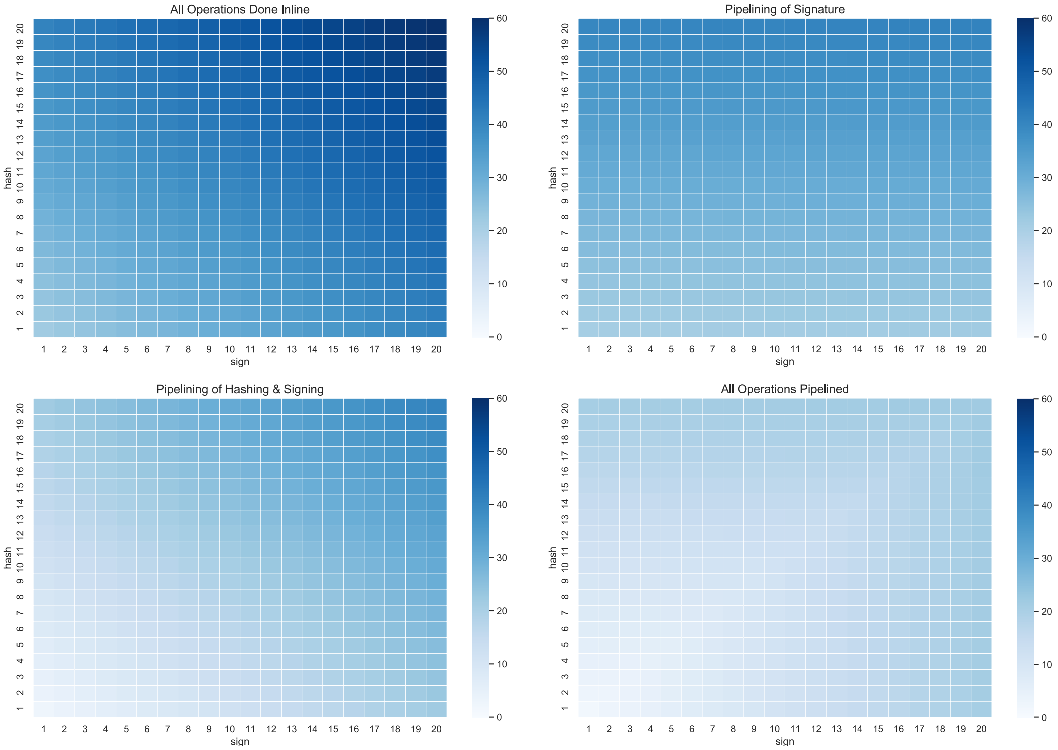


Figure 35: Transaction performance comparison heatmap

References

[1] “Amazon QLDB,” *Amazon Web Services, Inc.* https://aws.amazon.com/qldb/ (accessed May 02, 2019).

[2] Apache Foundation, “OpenAPI Specification v3.1.0 | Introduction, Definitions, & More.” https://spec.openapis.org/oas/v3.1.0 (accessed Jan. 21, 2022).

[3] “API Specifications Conference,” *Linux Foundation Events*. https://events.linuxfoundation.org/openapi-asc/ (accessed Jan. 21, 2022).

[4] “BigchainDB 2.0 Whitepaper • • BigchainDB,” *BigchainDB*. https://www.bigchaindb.com/whitepaper/ (accessed May 11, 2019).

[5] “Canonical’s Snap: The Good, the Bad and the Ugly,” *The New Stack*, Jul. 07, 2016. https://thenewstack.io/canonicals-snap-great-good-bad-ugly/ (accessed Aug. 12, 2020).

[6] “Cost of Cibercrime - Accenture.” Accessed: Nov. 07, 2019. [Online]. Available: https://www.accenture. com/_acnmedia/pdf-96/accenture-2019-cost-of-cybercrime-

study-final.pdf.

[7] "Designing Better File Organization Around Tags, Not Hierarchies," https://www.nayuki.io/page/designing-better-file-organization-around-tags-not-hierarchies#git-version-control (accessed Oct. 12, 2019).

[8] G. W. Dunlap, S. T. King, S. Cinar, M. A. Basrai, and P. M. Chen, "ReVirt: Enabling Intrusion Analysis through Virtual-Machine Logging and Replay," p. 14, 2002.

[9] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, Irvine, CA, 2000.

[10] A. Goel, Wu-chang Feng, D. Maier, Wu-chi Feng, and J. Walpole, "Forensix: A Robust, High-Performance Reconstruction System," *25th IEEE International Conference on Distributed Computing Systems Workshops*, Columbus, OH, USA, pp. 155-162, 2005. doi: 10.1109/ICDCSW.2005.62.

[11] "Gramm-Leach-Bliley Act," *Federal Trade Commission*. https://www.ftc.gov/tips-advice/business-center/privacy-and-security/gramm-leach-bliley-act (accessed Oct. 12, 2019).

[12] R. Hasan, R. Sion, and M. Winslett, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance," *Proccedings of the 7th Conference on File and Storage Technologies*, Berkeley, CA, USA, pp. 1-14, 2009. Accessed: Oct. 11, 2019. [Online]. Available: http://dl.acm.org/citation.cfm?id=1525908.1525909

[13] I. Khalil, S. El-Kassas, and K. Sobh, "DBKnot: A Transparent and Seamless, Pluggable, Tamper Evident Database," *EPiC Series in Computing*, 77:90-103, Oct. 2021. doi: 10.29007/7l81.

[14] L. Lavaire, "Immutable Systems: How They Work and Why Should We Care," *Medium*, Jul. 10, 2019. https://medium.com/nitrux/immutable-systems-how they-work-and-why-should-we-care-39e567a59f28 (accessed Aug. 12, 2020).

[15] "MD5, SHA-1, SHA-256 and SHA-512 Speed Performance – Automation Rhapsody." https://automationrhapsody.com/md5-sha-1-sha-256-sha-512-speed-performance/ (accessed Aug. 23, 2020).

[16] "Object-relational Mappers (ORMs)." https://www.fullstackpython.com/object-relational-mappers-orms.html (accessed Aug. 23, 2020).

[17] "Object-Relational Mapping," *Wikipedia*. Aug. 20, 2020. Accessed: Aug. 23, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Object-relational_mapping&oldid=974070664

[18] O. for C. Rights (OCR), "Summary of the HIPAA Security Rule," *HHS.gov*, Nov. 20, 2009. https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index. html (accessed Oct. 12, 2019).

[19] "OpenAPI Specification," *Wikipedia*. Nov. 27, 2021. Accessed: Jan. 21, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=OpenAPI_Specification&oldid=1057453112

[20] "OSTree." https://ostree.readthedocs.io/en/latest/ (accessed Aug. 12, 2020).

[21] "Overview of RESTful API Description Languages," *Wikipedia*. Jan. 17, 2022. Accessed: Jan. 21, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Overview_of_RESTful_API_Description_Languages&oldid=1066320532

[22] M. G. Oxley, "H.R.3763 - 107th Congress (2001-2002): Sarbanes-Oxley Act of 2002," Jul. 30, 2002. https://www.congress.gov/bill/107th-congress/house-bill/3763 (accessed Oct. 12, 2019).

[23] K. E. Pavlou and R. T. Snodgrass, "Forensic Analysis of Database Tampering," *ACM Trans Database Syst*, 33(4)30:1-30:47, Dec. 2008, doi: 10.1145/1412331.1412342.

[24] K. Pavlou and R. Snodgrass, "DRAGOON: An Information Accountability System for High-Performance Databases," *Proc. - Int. Conf. Data Eng.*, pp. 1329-1332, Apr. 2012. doi: 10.1109/ICDE.2012.139.

[25] "RAML (Software)," *Wikipedia*. Oct. 14, 2021. Accessed: Jan. 21, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=RAML_(software)&oldid=1049813969

[26] "Report to the Nations - 2018 Global Study on Occupational Fraud and Abuse," Association of Certified Fraud Examiners, 2019. Accessed: Apr. 17, 2019. [Online]. Available: https://www.acfe.com/report-to-the-nations/behind-the-numbers/

[27] "Representational State Transfer," *Wikipedia*. Jan. 06, 2022. Accessed: Jan. 21, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=1064071285

[28] "RFC 4810 - Long-Term Archive Service Requirements." https://datatracker.ietf.org/doc/rfc4810/ (accessed May 01, 2019).

[29] "Security by Design Principles - OWASP." https://www.owasp.org/index.php/Security_by_Design_Principles (accessed May 01, 2019).

[30] "Snapcraft - Snaps Are Universal Linux Packages," *Snapcraft*. https://snapcraft.io/ (accessed Aug. 12, 2020).

[31] D. M. Upton and S. Creese, "The Danger from Within," *Harvard Business Review*, no. September 2014, Sep. 01, 2014. Accessed: May 09, 2019. [Online]. Available: https://hbr.org/2014/09/the-danger-from-within

[32] S. Watts, "REST vs CRUD: Explaining REST & CRUD Operations," *BMC Blogs*. https://www.bmc.com/blogs/rest-vs-crud-whats-the-difference/ (accessed Jan. 21, 2022).

[33] "Welcome," *RAML*. https://raml.org/ (accessed Jan. 21, 2022).

[34] "Welcome to Flatpak's Documentation! — Flatpak Documentation." https://docs.flatpak.org/en/latest/ (accessed Aug. 12, 2020).

[35] Weltwirtschaftsforum and Zurich Insurance Group, *Global risks 2019: insight report*. 2019. Accessed: Nov. 07, 2019. [Online]. Available: http://www3.weforum.org/docs/ WEF_Global_Risks_Report_2019.pdf

[36] "What is Object/Relational Mapping? - Hibernate ORM." https://hibernate.org/orm/what-is-an-orm/ (accessed Aug.

23, 2020).

[37] "What is REST." https://restfulapi.net/ (accessed Jul. 26, 2020).

[38] C. Xia, G. Yu, and M. Tang, "Efficient Implement of ORM (Object/Relational Mapping) Use in J2EE Framework: Hibernate," pp. 1–3, Jan. 2010, doi: 10.1109/CISE.2009.5365905.

[39] K. Zeng, "Publicly Verifiable Remote Data Integrity," *Information and Communications Security*, pp. 419-434, 2008.

Islam Khalil has received his BSc and MSc degrees in computer science from The American University in Cairo. He is currently pursuing his PhD with a primary focus on database systems and security. On the professional side, Khalil is the co-founder of companies that provide data analytics, business intelligence, and cloud services for various enterprises in the areas of telecommunication, AgTech, utilities, defense, and others worldwide. Khalil's company has been recognized as one of the top 5 companies worldwide in applying artificial intelligence to the field of agriculture and one of the top 3 worldwide most influential companies in data analytics in some specific agriculture verticals. Khalil has been appointed by the minister of industry on the board of directors of various semi-governmental organizations focused on industry and export development.

Sherif El-Kassas is a Professor of Computer Science and Engineering at the American University in Cairo. El-Kassas' research interests are focused on Security Engineering, the application of formal methods in Software engineering and Computer Security, and Open Source technologies.

El-Kassas is also a consultant for various organizations; Member of the board of e-finance (leading provider of governmental and payment services), former board member of the Information Technology Industry Development Agency (ITIDA); Member of the board of trustees of the Egyptian e-signature center of excellence; Founding member of the Egyptian Open Source NGO (OpenEgpt) and Internet Masr (Egyptian Chapter of Internet Society); Founding partner, former broad member, and former CTO of SecureMisr (leading Egyptian Information Security services providers, recently acquired Cysiv a trend micro company); Founder of new startup, QuiverLabs, focusing on innovative threat modeling and incident response technologies; and Member of various professional computing societies.

El-Kassas received his Ph.D. from the Eindhoven University of Technology in the Netherlands.

Karim Sobh received the B.Sc., M.Sc., and Ph.D. degrees in computer science from The American University in Cairo. He worked at the American University in Cairo (AUC) as a Full time Assistant Professor for three academic years in the Department of Computer Science and Engineering. Prior to that he worked at Nile University (NU), Cairo and as an Assistant Professor and the University of California at Santa Cruz (UCSC) as a Visiting Lecturer. He is currently the Chief Technology Officer (CTO) of Blnk Consumer Finance, an emerging FinTech startup in Egypt. He also founded Code-Corner, a software development firm providing software development, subcontracted services, cloud deployment services, consultation services, and turn-key solutions using open source technologies. He also worked as a Systems Architecture Consultant at IBM Egypt. His role included and was not limited to providing system architecture consultations and implementation services for large projects. His specialization is in operating systems, networks, distributed systems, and cloud computing, and his Ph.D. topic is cloud environments metering.

Design and Implementation of VS-TAP The Veteran Services Tracking and Analytics Program

Jonathon Hewitt*, Daniel Hall*, Christopher Parks*, Payton Knoch*,
Sergiu M. Dascalu*, Devrin Lee*, Nikkolas J. Irwin*, Frederick C. Harris, Jr.*
University of Nevada, Reno,
Reno, Nevada, USA.

Abstract

The Veteran Services Tracking and Analytics Program (VS-TAP) is a web application used to store and query the rate and duration of visitors within Veteran Services' locations. The application accepts data from Navigate as well as a hosted demographics survey to display statistics in a graphically meaningful way. Accumulating data from different sources allows stakeholders to create custom reports to compare multiple variables that represent student veterans.

Key Words: Analytics, authentication, data, database, django, document processing, ETL (extract, transform, load), systemd-nspawn, tracking, veteran services, visualization, web application.

1 Introduction

The Veteran Services Tracking and Analytics Program (VS-TAP) is a data gathering and analytics application. The goal of this program is to collect, store, and combine data from several sources into a single usable database. The web application tracks the rate and duration of visitors that attend veteran centers and events. The program also combines all the data collected from various sources that can be queried for data visualization purposes. Data capture and visualization are important to the center's existence and helps determine the success of events as well as requests for funding.

The interface for data visualization is presented as a "reports wizard" to help walk Veteran Services staff through graph generation. The initial aim was to mimic the quantity of graphs associated with Microsoft Excel while eliminating the learning curve. Previously, Veteran Services manually tracked attendance using a USB-connected barcode scanner. Veteran Services staff were unable to obtain demographic information directly from the barcode scanner. After tracking attendance with the barcode scanner during a given time frame, staff members would periodically send the data containing student information to the Office of Data Analytics. The staff at the Office of Data Analytics would match the demographics with the student barcode information and send an Excel report back to Veteran Services. The Excel sheet would display demographic information for each associated student barcode entry.

Veteran Services also collected additional demographics that were not available from the Office of Data Analytics. Veteran Services used a custom Google survey from an iPad device. First-time visitors would fill out the survey on the iPad upon entry into the facility. Staff members would periodically export the survey data via an Excel spreadsheet. Staff members would have a total of three spreadsheets to build reports: Attendance in/out information, demographics provided by the Office of Data Analytics, and the Google survey demographics. Using the three spreadsheets, staff members would manually reconcile and match student data to build the reports using chart wizards provided by Microsoft.

During the development of VS-TAP, VS implemented an upgrade to the barcode scanner system. VS implemented a student identification (WolfCard) scanner. The upgraded scanner is able to scan student ID cards and allow Veteran Services staff direct access to demographic information instead of obtaining this information from the Office of Data Analytics. VS staff continues to use a survey to obtain supplemental demographic information. VS-TAP includes the built-in implementation of the survey that directly feeds survey data into the database, instead of using a Google Forms survey. VS-TAP aims to allow staff members to upload only two spreadsheets that are automatically parsed and updated into the database. The staff can then use a reports wizard to obtain the appropriate charts and tables. The reports wizard was designed to give staff members more control over graph axis, titles, and graph aesthetics than was previously possible using Microsoft excel.

Concerning security, VS-TAP was designed to protect against malicious actors. To this extent developers integrated user authentication, protection from SQL injections to the database, as well as CSRF (Cross-Site Request Forgery) token validation. In addition to the security mentioned, VS-TAP is only accessible from the University of Nevada, Reno (UNR) network to limit external network traffic.

The VS-TAP web application was designed to be containerized using systemd-nspawn [5] which is native to the Linux operating system. In May 2021, VS-TAP was launched on the College of Engineering's virtual server at the University of Nevada, Reno.

The rest of this paper is structured as follows: Section 2

*Department of Computer Science and Engineering. 1664 North Virginia Street / MS 171. Reno, NV 89557. Email: {jonathonhewitt, danielhall, christopherparks, pknoch}@nevada.unr.edu, dascalus@cse.unr.edu, dllee@unr.edu, nikkolasjirwin@nevada.unr.edu, fred.harris@cse.unr.edu

presents the motivation and design of VS-TAP including functional and non-functional requirements as well as the application's use cases. Section 4 covers the technologies used to implement the current version of VS-TAP. The final version of VS-TAP along with screen shots are given in Section 5. VS-TAP conclusion as well as future works are given in Section 6.

2 Motivation and Design

Manually collecting visit data is difficult and unreliable, and the kinds of reports you can generate from this data is limited. By automating the check-in and check-out procedures at the Veteran Services offices and collecting data in the process, the amount of useful reports that can be created increases. The main goals for this project is to provide a seamless check-in and check-out experience and to augment the kinds of reports that can be generated. To make sure these goals are adequately met, a list of functional and non-functional requirements are created alongside a list of desired use cases.

2.1 Similar Applications

Data analytics are commonly used across multiple industries. Tablaeu, part of Salesforce's software suite, allows organizations to analyze and visualize data from multiple sources that is fed into a single platform [10]. For example, users of Tableau can use data from sales, marketing, and business expenses to generate detailed, visual reports [10]. VS-TAP provides a similar concept - using a central location for importing and visualizing data. The difference between VS-TAP and Tableau is that VS-TAP is more specialized for attendance tracking that is build to incorporate the specific third-party technology that is already used at Veteran Services.

Attendance tracking is commonly used among businesses for hourly employees. Kronos is a timekeeping software that is used to track employee attendance, employee time off and vacation, help businesses with remaining compliant with labor regulations, and provide detailed visualizations [4]. Kronos software allows businesses to obtain detailed demographic information based on employee attendance, such as employee count by state, comparing shift hours worked against shift hours scheduled, and employee headcount by business location [3]. Kronos is the most similar software to VS-TAP in that it is primarily used for tracking attendance. Kronos tracks existing employees that are in regular attendance. While VS-TAP has visitors in regular attendance, VS-TAP is meant to handle new visitors on a daily-basis with the integrated survey. Additionally, VS-TAP is meant to provide a lower learning curve for its targeted users.

Microsoft Excel is another tool used for tracking attendance and generating reports. Excel allows users to manually enter data into tabular format, known as a ”spreadsheet” [6]. Prior to the development of VS-TAP, Microsoft Excel was used by staff members. Excel requires its users to manually filter out unnecessary or repetitive data that is not used in the visual reports. Excel also allows its users to build charts by selecting the relevant data entries and choosing from multiple options in a wizard. Additionally, if multiple spreadsheets are used in a single chart, it requires its users to combine the spreadsheets. Filtering out data and combining the spreadsheets can take up to several hours. Although VS-TAP still involves Excel spreadsheets, it automates the data selection for report generation based on user criteria. Furthermore, VS-TAP automates the process of filtering and combining spreadsheets.

2.2 Functional Requirements

Functional requirements, per Ian Sommerville [9], are used to describe the necessary functionality of a system. These requirements are directly seen in the final project. The following is a list of functional requirements for the VS-TAP system.

The System shall:

1. Parse scanner data from Navigate.
2. Store visit and demographic data in a database.
3. Allow users to query the database for data reports and display on the reports page.
4. Allow users to specify events for visit data.
5. Allow users to create an account.
6. Allow users to log in to their accounts.
7. Implement a navigation page that links each page on the site.
8. Allow users to export reports as images for reports.
9. Allow users to export report tables as CSV files.
10. Allow users to search individual students.
11. Allow users to remove individual students from the visit data.
12. Display different home pages for authorized and unauthorized visitors.
13. Provide a wizard as a user interface for creating new reports.
14. Allow users to specify a range of dates for reporting.
15. Allow users to change their password.
16. Allow users to upload a profile picture associated with their account.
17. Allow users to change their account first and last name.
18. Allow users to change their email address.
19. Support manual upload of visits when scanners are unavailable.
20. Allow users to quickly query for individual statistics e.g. average visit duration.
21. Allow users to save templates for data visualizations and load them with new data points.
22. Provide an administrative page for managing all user accounts.
23. Allow administrators to change names, email addresses, passwords, and profile pictures of other users within the system.

24. Allow users to change the name of each saved report type.
25. Provide a dynamic wizard page for adding stacked graphs.
26. Allow users to download reports as PDF files.
27. In addition to the wizard, provide an interactive dashboard for quickly creating new reports.
28. Automatically import visit data from Navigate on a live basis.
29. Provide a portal for quickly sharing visit data to other users.

2.3 Non-Functional Requirements

Non-functional requirements, per Ian Sommerville [9], are used to describe the quality constraints that a system must satisfy. The following is a list of non-functional requirements for the VS-TAP system.

1. The site will be hosted and run on the UNR network.
2. Allow for multiple concurrent users to upload data and create visualizations
3. The site should return queries for data, and data visualizations quickly
4. The site should be easy to navigate for people with little to no technical knowledge
5. The data reporting options should be shown in a straightforward and usable manner
6. The site should have minimal downtime
7. The site should be robust to bad data uploads
8. The site should be non portable and only accessible from the campus network
9. Users should be able to obtain all visual reports that are needed for funding of VS
10. All information protected by FERPA must be secure from unauthorized access
11. The software should be designed in a way that does not need frequent updates
12. The code should be easily maintainable in case future updates to the software are necessary
13. The software should function offline during downtime

2.4 Detailed Use Cases

This subsection presents the detailed use cases. Figure 1 gives the use case diagram.

- **AccountLogin:** When the user first enters the website, the user will be prompted for a user name and password. If the credentials are correct, the user will be taken to the home page.
- **ChangePassword:** If the user wants to change their password, they can select `Change Password`. The user will be prompted for their current password, the new password, and a second entry for their new password. The current password must be correct and the two entries for the two passwords must match. If all forms are correct, the user will receive a message that their password was successfully changed. If the current password is incorrect or the two fields for the new password do not match, an error message will display.
- **SelectReportPage:** When the user selects `Visualizations` from the navigation bar or enters the “Visualizations” view from the address bar, the user will be taken to the visualizations page.
- **SelectImportPage:** When the user selects `Upload Files` from the navigation bar or enters the “Import” view from the address bar, the user will be taken to the upload page.
- **SelectHomePage:** When the user selects `Home` from the navigation bar or enters the “Home” view from the address bar, the user will be taken to the home page.
- **ImportFile:** On the **Import** page, the system will prompt the user for a file. The user will select the file from their computer. If the file is successfully uploaded to the server, the system will indicate to the user that it was successful; otherwise an error message will display. After a successful upload, the parser will begin parsing the file.
- **GetIndividualStatistic:** On the **Reports** page, one of the options that the system provides to the user is the ability to select an individual statistic (e.g. average G.P.A, total number of visitors on 10/31/2020). The user will select from the available individual statistics, then click `Get Individual Statistic` to obtain the statistical report.
- **DownloadFile:** On the **Reports** page, the user will have the option to download any data visualization that they select to their computer. For example, if they select a bar graph, they can download the graph as an image.
- **PlotData:** On the **Reports** page, the user will select from a list of options for a specific type of graph. After selecting the options, the user will click a button that will submit a user request to the system to plot the data. The visualizations module should return the plotted data to the user in the form of a graph.
- **CompleteSurvey:** Upon the first visit of the VS office, the student is taken to the **Survey** page to complete a list of fields.
- **SubmitSurvey:** Upon completing the survey, the student clicks `Submit`. If all fields are correctly filled out, the survey data is inserted into the database; otherwise, an error message appears.
- **GetReport:** After filling out the reports wizard, users can get a detailed report about attendance for a given data range, including both visual and tabular data.
- **SelectPreset:** After selecting a specific saved report from the list of saved reports, a user is given the details of that report with options, such as creating a new report from the saved report preset and deleting the preset altogether.
- **SavePreset:** After obtaining a report from the Reports Wizard, the user is given the option to save the preset. If the user saves the preset, the preset is saved into the database where the use can access it via the **Presets** page.

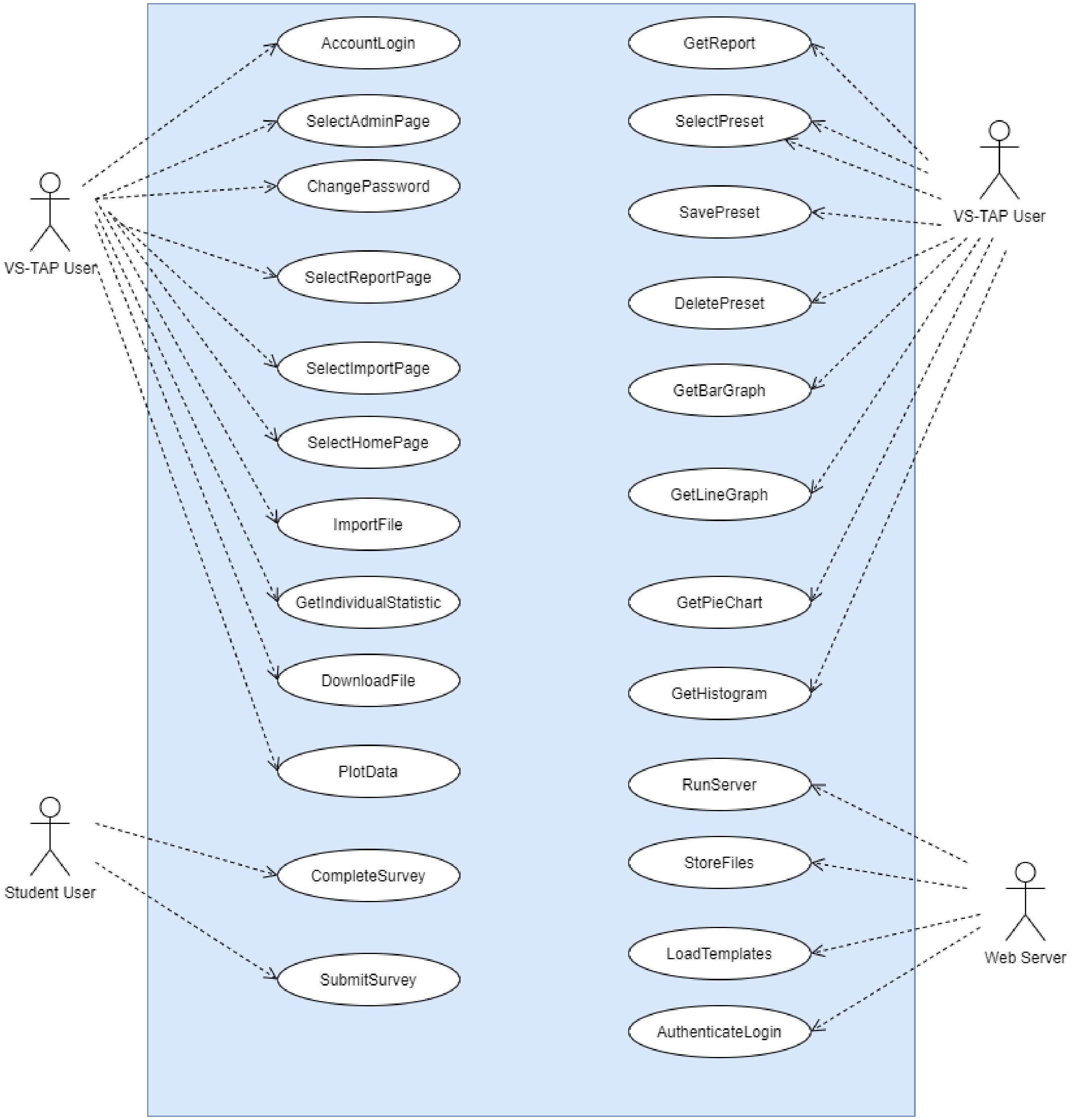


Figure 1: Use case diagram

- **DeletePreset:** The report preset is deleted from the database after the user selects `Delete Preset` and the preset no longer appears in the list of saved presets.
- **GetBarGraph:** In the reports wizard, the user obtains visit, demographic, and/or survey data in bar graph format.
- **GetLineGraph:** In the reports wizard, the user obtains visit, demographic, and/or survey data in line graph format.
- **GetPieChart:** In the reports wizard, the user obtains visit, demographic, and/or survey data in pie chart format.
- **GetHistogram:** In the reports wizard, the user obtains visit, demographic, and/or survey data in histogram format.
- **RunServer:** Upon execution of `manage.py`, the web server loads the software and makes it available to its users.
- **StoreFiles:** The web server will maintain storage of all files, including database files and user profile pictures.
- **LoadTemplates:** In conjunction with Django, the web server is responsible for loading all template (HTML) files that will display web page content to the end user.
- **AuthenticateLogin:** The web server should authenticate the user's credentials when they try to log into the systems. If the password or username are not correct, the server shall deny user access to the system.

3 Acceptance Criteria and Testing Strategy

3.1 User Stories

User stories are used to verify that the application meets the usability requirements for the end user. The development team worked closely with the employees at the Veteran Services center to come up with a list of user stories that can be broken up into discrete tasks which can be independently tested and implemented. Below is a list of user stories for the VS-TAP application.

- As a user with an existing account I want to be able to log in so that I can create and view reports.
  - When users input valid credentials, they are logged in to the appropriate account.
  - After logging in the user is given access to create reports.
  - User profile settings are stored to their account.
  - Users should be able to input their credentials into text forms.
- As a user without an account I want to be able to create an account so that I can login in the future.
  - Users without accounts can create an account by visiting the create account page.
  - User login credentials are stored in the database, allowing users to login after creating.
  - Users can enter credentials into text forms to create account.
- As a user I want to be able to upload .csv files with visit data so that I can use that data in future reports.
  - Users can visit an upload page and upload a document.
  - If the document is a .csv file with navigate data, it is parsed into a list of visits.
  - Parsed data is inserted into the database so that it can be queried later.
- As a user I want to be able to manually input visit data for visitors without Wolf Cards.
  - Users can visit a manual entry form page.
  - Users can input and submit visit data in a series of text fields.
  - When submitted, the manually entered data is inserted into the database so that it can be queried later.
- As a user I want to be able to create dynamic visualizations to reflect visitor trends.
  - Users can visit the custom reports page.
  - Users are walked through a creation wizard process for data visualization.
  - Users are given the option to save customization fields for reuse.
  - After finalization, a table and graph matching user specifications is generated.
  - Users may download visualized report.
- As an administrative user I want to be able to create accounts for other users.
  - Admins can visit a user creation page.
  - Admins can input account credentials to create an account for other users.
  - The new user credentials are stored into the database.
  - After new account creation, the new user can log in and view the site.
- As an administrative user I want to be able to delete other user accounts.
  - Admins can visit an account list page.
  - From the account list page, users can select individual user accounts.
  - From an individual user account profile, admins can select to delete an account.
  - If an account is deleted, it's entry in the database is removed.
  - After account deletion, that user can no longer log in and view the site.
- As an administrative user I want to be able to edit other user's login credentials and personal information.
  - Admins can visit an account list page.
  - From the account list page, users can select individual user accounts.
  - From an individual user account profile, admins can select to edit an account.

  - If an account is selected to be edited, a set of text forms are presented.
  - If an admin alters the data in the text field from the user's current settings the new information replaces the old field in the database.
  - If a user's login credentials are changed, that user can no longer log in with their old credentials.
- As a user I want to be able to create a bar graph demonstrating the number of visits for each day in a month.
  - When visiting the visualization page, users can select bar graph as an option to generate.
  - After selecting bar graph, users can select usage by date as an option to graph.
  - Users can manually set the colors and scaling for the bar graph.
  - Users can manually chose a range of dates to pull data from.
  - After users select all of the relevant options, they can chose to view the report.
  - Data should be pulled from the data base to generate the report.
  - After selecting to view the report, users are shown a bar graph demonstrating the number of visits for each day in their date range.
- As a user I want to be able to export reports as images and .csv files to include in other files.
  - When generating a report, users should have the option to include a data table in the report.
  - When viewing a report, users should have the option to export each figure as a .png they can download.
  - When viewing a report, if a data table was included, users should have the option to export the table as a .png they can download.
  - When viewing a report, if a data table was included, users should have the option to export the table as a .csv file they can download.
- As a user I want to be able to select a page from a navigation bar so that I can easily change between different pages on the website.
  - A navigation bar with a list of available pages should be visible to users at all times.
  - When clicking on a page from the navigation bar users should be taken to the selected page.
  - When a user clicks on a different page while filling in forms in another page those forms are discarded.
- As a user I want to be able to select log out from the navigation bar so that I can log out and exit the site.
  - Log out should be an option on the navigation bar.
  - When log out is selected the user is taken to the login splash page.
  - If a user is logged out, they have to re-enter their credentials to enter the site.

3.2 Testing Strategy

The benefit of working closely with the project's end users is the efficacy of acceptance and user tests. The VS-TAP team is able to host a project built for the stakeholders so that they can use it and report any bugs or underdeveloped features. These testing strategies are the main strategies employed to test user experience, while automated testing is used to verify that each page of the web-app is accessible. Table 1 outlines some of the testing done.

The Test Type column indicates what category of test that test falls under. The two main categories are automated tests, which are tests that are run programmatically and user and acceptance tests, which involve the end users using the product to make sure it meets specifications. The Target File or Screen column indicates what part of the project is being tested by the test. The Test Data or Situation indicates what environment the project is being tested under. Lastly, the Outcome and Actions Required column indicates what was found and needed to be improved as a result of the testing.

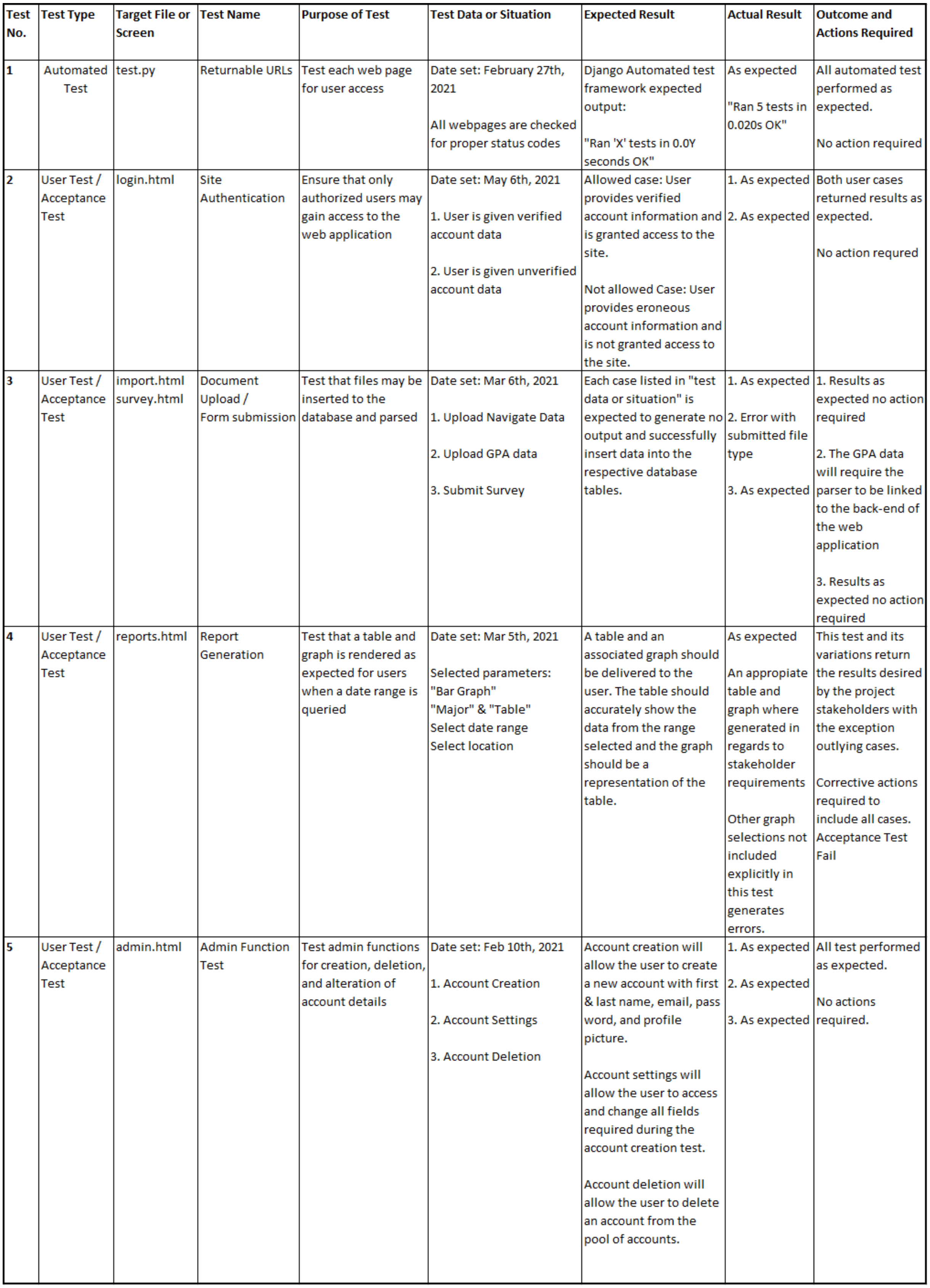
4 Technologies Used

VS-TAP uses Django as the main architecture. As VS-TAP is a web application, the frontend features Hypertext Markup Language (HTML), Cascading Stylesheets (CSS) to provide visual enchancements to the object displayed via HTML, and JavaScript to provide any interactivity to the users. The backend logic is handled via Python scripts. SQLite3 is the database containing all of the visit and demographic data. Additionally, the team used the Bootstrap HTML library for faster frontend development time and JQuery for handling user events in JavaScript.

Django: Django [2] uses a concept known as Model-View-Template (MVT), which is based off of the Model-View-Controller architectural pattern. Django models feature objects that interact with the integrated database, such as SQLite3. A model object contains all of the database fields associated with the object. Each instance of the object corresponds to an entry in the database. The View contains all of the functions that render the HTML templates and the associated logic performed prior to the rendering. The Template is the HTML templates that are displayed, including any accompanying JavaScript or CSS styling.

HTML/CSS/JavaScript: A majority of the frontend starts with a base HTML page that contains the common styling and layout used for all of the web pages. Specific web pages (e.g. Reports page) extend from the base HTML page. Django provide dynamic elements in the HTML pages through the use of context variables. A context variable is a variable whose value is calculated by the backend via Python functions. The output varies based on conditions such as the database contents and user input. CSS provides styling to individual HTML

Table 1: Acceptance Test Plan



Test No.	Test Type	Target File or Screen	Test Name	Purpose of Test	Test Data or Situation	Expected Result	Actual Result	Outcome and Actions Required
1	Automated Test	test.py	Returnable URLs	Test each web page for user access	Date set: February 27th, 2021 All webpages are checked for proper status codes	Django Automated test framework expected output: "Ran 'X' tests in 0.0Y seconds OK"	As expected "Ran 5 tests in 0.020s OK"	All automated test performed as expected. No action required
2	User Test / Acceptance Test	login.html	Site Authentication	Ensure that only authorized users may gain access to the web application	Date set: May 6th, 2021 1. User is given verified account data 2. User is given unverified account data	Allowed case: User provides verified account information and is granted access to the site. Not allowed Case: User provides eroneous account information and is not granted access to the site.	1. As expected 2. As expected	Both user cases returned results as expected. No action requred
3	User Test / Acceptance Test	import.html survey.html	Document Upload / Form submission	Test that files may be inserted to the database and parsed	Date set: Mar 6th, 2021 1. Upload Navigate Data 2. Upload GPA data 3. Submit Survey	Each case listed in "test data or situation" is expected to generate no output and successfully insert data into the respective database tables.	1. As expected 2. Error with submitted file type 3. As expected	1. Results as expected no action required 2. The GPA data will require the parser to be linked to the back-end of the web application 3. Results as expected no action required
4	User Test / Acceptance Test	reports.html	Report Generation	Test that a table and graph is rendered as expected for users when a date range is queried	Date set: Mar 5th, 2021 Selected parameters: "Bar Graph" "Major" & "Table" Select date range Select location	A table and an associated graph should be delivered to the user. The table should accurately show the data from the range selected and the graph should be a representation of the table.	As expected An appropiate table and graph where generated in regards to stakeholder requirements Other graph selections not included explicitly in this test generates errors.	This test and its variations return the results desired by the project stakeholders with the exception outlying cases. Corrective actions required to include all cases. Acceptance Test Fail
5	User Test / Acceptance Test	admin.html	Admin Function Test	Test admin functions for creation, deletion, and alteration of account details	Date set: Feb 10th, 2021 1. Account Creation 2. Account Settings 3. Account Deletion	Account creation will allow the user to create a new account with first & last name, email, pass word, and profile picture. Account settings will allow the user to access and change all fields required during the account creation test. Account deletion will allow the user to delete an account from the pool of accounts.	1. As expected 2. As expected 3. As expected	All test performed as expected. No actions required.

elements, classes of HTML elements, or entire pages. CSS style options include (but are not limited to) centering, changing the color, font size, and font color. JavaScript provides interactivity to the page based on user actions, such as clicking on a button and typing in a text entry.

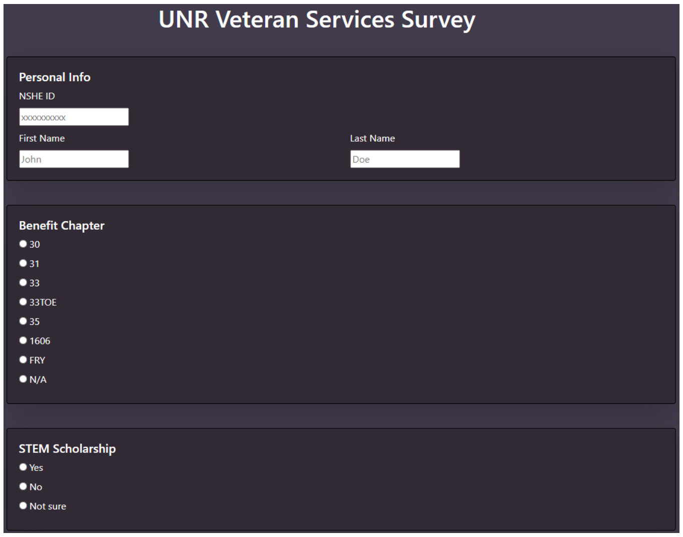
Dash: Dash is a library used for creating detailed and informative interfaces that provide visual reports through Plotly [7]. VS-TAP uses Dash because each visit report needs visual representation, such as a Bar Graph or a Pie Chart. Django contains an extension known as Django-Plotly-Dash. Django-Plotly-Dash provides tools for integrating Dash components within the Python scripts and HTML code. The backend of the reports page is written using Plotly functions and variables in Python while the graph itself is rendered by Dash.

SQLite3: SQLite3 is used for the database. The database logic for the user accounts and the report presets is automatically carried out by Django models. The logic for the student demographics and visits are direclty handled by SQLite3 commands embedded in the Python view functions within the parser and the reports modules.

5 Results

Aside from security, such as user authentication, there are three main sections of the VS-TAP web application that Veteran Services' staff and visitors interact with: the student survey, the document upload section, and the data visualization page.

Student Survey: The student survey is a multiple choice questionnaire which is accessible by students using a QR code located within each veteran center. Data collected from the student survey helps Veteran Services determine relationships between various demographic data, student's involvement in the center, and student's academic performance. Figure 2 and 3 show the beginning and end of an 18 question survey that helps the VS-TAP web application create more dynamic reports to better serve student veterans.



UNR Veteran Services Survey

Personal Info

NSHE ID: xxxxxxxxxx

First Name: John

Last Name: Doe

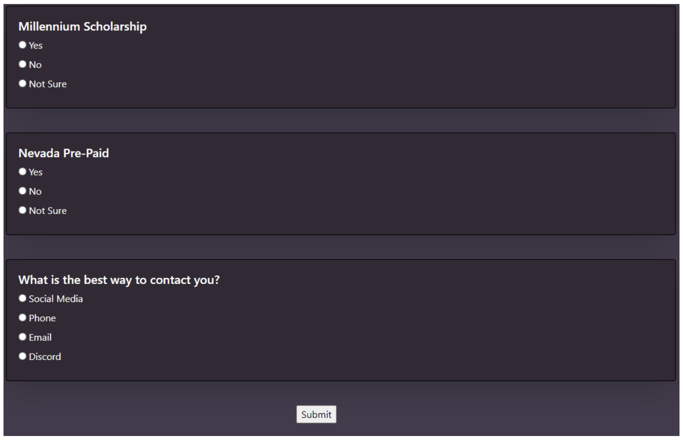
Benefit Chapter

- ● 30
- ● 31
- ● 33
- ● 33TOE
- ● 35
- ● 1606
- ● FRY
- ● N/A

STEM Scholarship

- ● Yes
- ● No
- ● Not sure

Figure 2: The UNR Veteran Services Survey requires each visitor's NSHE ID to relate their survey demographics to their visit data



Millennium Scholarship

- ● Yes
- ● No
- ● Not Sure

Nevada Pre-Paid

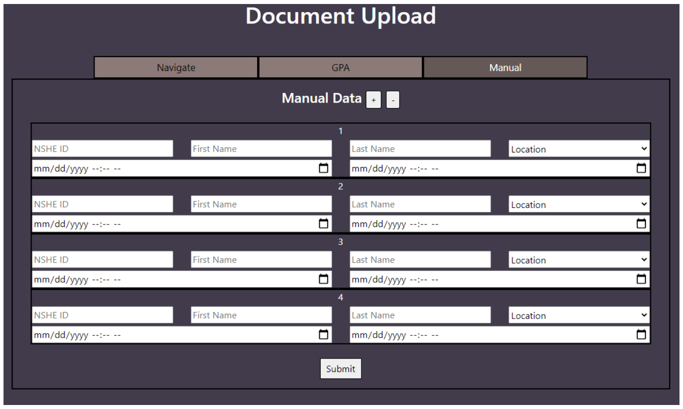
- ● Yes
- ● No
- ● Not Sure

What is the best way to contact you?

- ● Social Media
- ● Phone
- ● Email
- ● Discord

Figure 3: When each survey is submitted the student's responses are stored along their visits data for future querying

Document Upload: The document upload section allows users to upload documentation from different sources. Once a document is submitted to the application, the web application extracts, transforms, and loads (ETL) the data into the back-end of the application. The document upload section allows the users to upload data from the university's Navigate system, GPA data, and manual entry data in the event that the Navigate system is down. Figure 4 show the document upload section



Document Upload

Manual Data

1			
NSHE ID	First Name	Last Name	Location
mm/dd/yyyy --:-- --		mm/dd/yyyy --:-- --	
2			
NSHE ID	First Name	Last Name	Location
mm/dd/yyyy --:-- --		mm/dd/yyyy --:-- --	
3			
NSHE ID	First Name	Last Name	Location
mm/dd/yyyy --:-- --		mm/dd/yyyy --:-- --	
4			
NSHE ID	First Name	Last Name	Location
mm/dd/yyyy --:-- --		mm/dd/yyyy --:-- --	

Figure 4: In the event that the university student tracking system is offline, staff at Veteran Services may still upload visits data manually

where users may upload two different comma separated value (csv) documents or enter manual student's visit data.

Data Visualization: The data visualization section walks the user through a "reports" wizard to create a graphical representation of specific visitor's data. Reports creation allows the user to generate various different graph types while querying 28 different student visitor parameters. Figure 5 and 6 show the results from one such query where the reports wizard creates a table and graph for the classification of students who visited the center between the dates of 04/05/2021-08/31/2021.

6 Conclusion and Future Work

The staff of Veteran Services (VS) can now use spreadsheets obtained from Navigate to upload them to VS-TAP. By uploading the spreadsheet data, the data is saved to a database and the data storage process is automated. Staff members of VS can specify the type of report, date range, and aesthetics to retrieve a report that is automatically generated. Previously, VS staff needed to manually inspect multiple spreadsheet pages to create a custom report for VS funding. VS-TAP has the potential to be used for other buildings at both the UNR campus and other universities. Other universities provide an equivalent to the Veteran Services building and may use a similar funding structure; therefore, it would be beneficial for other universities to use this software to track attendance.

Although there exists commercial software that tracks attendance, such as ADP [1], the commercial software is primarily concerned with tracking employee attendance for payroll purposes. VS-TAP customizes the attendance tracking to provide data visualizations and reports to obtain funding. In the future, VS-TAP's functionality can also be expanded to a more interactive dashboard of data visualizations. Currently, reports are generated by selecting from a list of customization options through a wizard format. If users can dynamically view how their customization choices can change the output of their reports, they can find their ideal customization settings at a faster rate. The interactive dashboard can be achieved through libraries such as Dash Enterprise [8].

VS-TAP requires users to upload visit data from a third party source. The third party source is Navigate. In the future, it would be beneficial if visit data reflected in real-time. Real-time visit data can be obtained through an auxiliary application within VS-TAP that uses hardware to scan the WolfCard, then uploads the visit to the VS-TAP database through a cloud infrastructure. By allowing real-time visit uploads, VS staff would be solely focused on obtaining the desired report needed for funding.

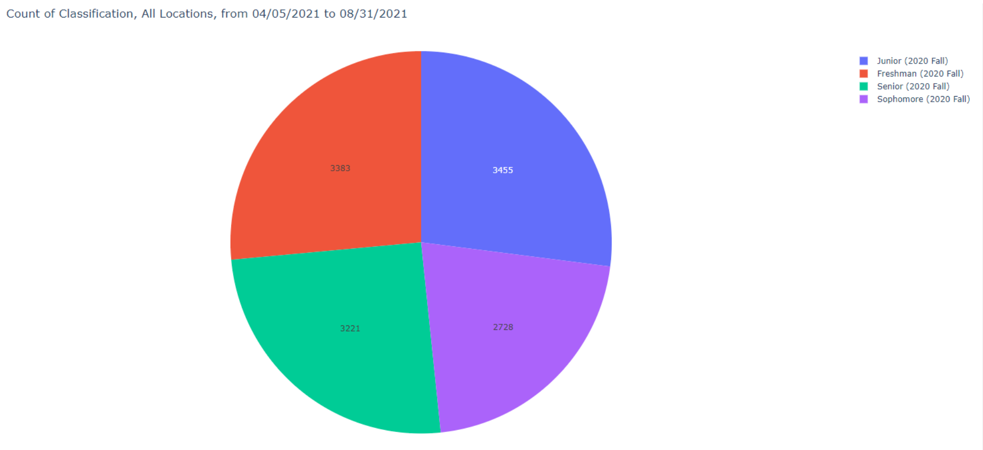
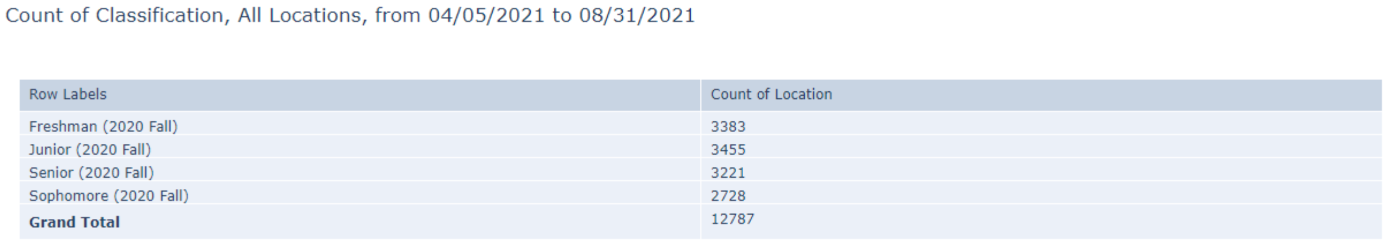


Figure 5: Pie chart representing the amount of students who visit the center and their associated classification year



Count of Classification, All Locations, from 04/05/2021 to 08/31/2021

Row Labels	Count of Location
Freshman (2020 Fall)	3383
Junior (2020 Fall)	3455
Senior (2020 Fall)	3221
Sophomore (2020 Fall)	2728
Grand Total	12787

Figure 6: Table representing the amount of students who visit the center and their associated classification year

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under grant number IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] ADP, Inc. "Employee Time Tracking". `https://www.adp.com/what-we-offer/time-and-attendance/employee-time-tracking.aspx` (Last Accessed: 2/3/2002).

[2] Django Software Foundation. "Django Documentation, Release 3.2.4.dev". `https://docs.djangoproject.com/en/3.2/#django-documentation` (Last Accessed: 2/3/2002).

[3] Kronos Incorporated. "Kronos Workforce Ready Data Visualizations". `https://www.kronos.com/resource/download/29126` (Last Accessed: 3/7/2002).

[4] Kronos Incorporated. "Time & Attendance System; Time Tracking Software — Kronos". `https://www.kronos.com/products/time-and-attendance` (Last Accessed: 3/7/2002).

[5] Linux.org. "Systemd-nspawn - Spawn a Namespace Container for Debugging, Testing and Building at Linux.org". `https://www.linux.org/docs/man1/systemd-nspawn.html` (Last Accessed: 2/3/2002).

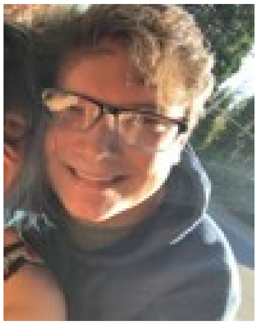
[6] Microsoft Corporation. "Microsoft Excel". `https://www.microsoft.com/en-us/microsoft-365/excel` (Last Accessed: 2/3/2002).

[7] Plotly. "Dash Documentation & User Guide — Plotly". `https://dash.plotly.com/` (Last Accessed: 2/3/2002).

[8] Plotly. "Dash Enterprise". `https://plotly.com/dash/` (Last Accessed: 2/3/2002).

[9] Ian Sommerville. *Software Engineering*. Pearson, 10th edition, 2016. `https://www.pearson.com/us/higher-education/program/Sommerville-Software-Engineering-10th-Edition/PGM35255.html` (Last Accessed: 2/3/2002).

[10] Tableau Software, LLC. "Salesforce + Tableau". `https://www.tableau.com/solutions/salesforce` (Last Accessed: 2/3/2002).



Jonathon Hewitt received his BS in Computer Science and Engineering from the University of Nevada, Reno in 2020. His research interests are in Software Engineering, Computer Graphics, Image Processing, and Security. Since he graduated he has pursued a career in Software Engineering and is currently working for a company that creates image processing and visualization tools for the medical and airport security industries.

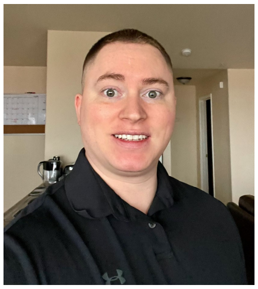
Daniel Hall is an alumni from the University of Nevada, Reno. He received his Bachelor of Science in Computer Science and Engineering in 2021. He also had a minor in Mathematics. His research interest are in Software Engineering, Game Design, Distributed Computing, and Big Data Systems. Since he graduated he has pursued a career in Software Engineering and is currently working for a defense contractor supporting the U.S. Dept. of Defense.

Christopher Parks is an alumni from the University of Nevada, Reno. He received his Bachelor of Science in Computer Science and Engineering in 2019. His research interest are in Software Engineering, Computer Graphics, and Cyber Security. Since he graduated he has pursued a career in Software Engineering and is currently working in a startup in the insurance industry.

Payton Knoch is an alumni from the University of Nevada, Reno. He received his Bachelor of Science in Computer Science and Engineering in 2019. His research interest are in the areas of Software Engineering, Video Games, and Cybersecurity. He has implemented projects to mimic an original network attack and defense model in cybersecurity and he has designed a video game on the Android operating system for personal entertainment. Payton has utilized his skills learned at the university to pursue a career as a software engineer in the insurance industry.

Sergiu M. Dascalu is a Professor in the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR), which he joined in July 2002. He received his PhD degree in Computer Science (2001) from Dalhousie University, Canada and a Master's degree in Automatic Control and Computers (1982) from the Polytechnic of Bucharest, Romania. At UNR he is also the Director of the Software Engineering Laboratory (SOELA) and the Co-Director of the Cyberinfrastructure Lab (CIL). Since joining UNR, he has worked on research projects funded by federal agencies (NSF, NASA, DoD-ONR) as well as the industry. He has advised 11 PhD and over 50 Master students. He received several awards, including the 2009 Nevada Center for Entrepreneurship Faculty Advisor Award, the 2011 UNR Outstanding Undergraduate Research Faculty Mentor Award, the 2011 UNR Donald Tibbitts Distinguished Teacher of the Year Award, the 2014 CoEN Faculty Excellence Award, and the 2019 UNR Vada Trimble Outstanding Graduate Mentor Award. He is a Senior Member of the ACM.

Devrin Lee finished her BS in Computer Science in 2005, and her MS in Information Systems in 2008 from the University of Nevada, Reno. She is a Project Management Professional and a certified ScrumMaster. She has done consulting for small businesses in the IT arena, was the manager of Technical Operations for PCLender, and is currently an Operational Program Manager for Microsoft. Her research interests are in software engineering, product design, and project management.



Nikkolas J. Irwin Nikkolas Irwin received his BS in Computer Science and Engineering from the University of Nevada, Reno in 2020. He currently works for the U.S. Department of Energy (DOE), Office of Inspector General (OIG), Office of Technology, Financial, and Analytics (OTFA) as a data scientist. His current interests include leveraging DataOps, MLOps, and more broadly DevOps to enhance data science workflows.

Frederick C. Harris, Jr. received his BS and MS degrees in Mathematics and Educational Administration from Bob Jones University, Greenville, SC, USA in 1986 and 1988 respectively. He then went on and received his MS and Ph.D. degrees in Computer Science from Clemson University, Clemson, SC, USA in 1991 and 1994 respectively.

He is currently a Professor in the Department of Computer Science and Engineering and the Director of the High Performance Computation and Visualization Lab at the University of Nevada, Reno. Since joining UNR, he has worked on research projects funded by federal agencies (NSF, NASA, DARPA, ONR, DoD) as well as industry. He is also the Nevada State EPSCoR Director and the Project Director for Nevada NSF EPSCoR. He has published more than 300 peer-reviewed journal and conference papers along with several book chapters and has edited or co-edited 14 books. He has had 14 PhD students and 81 MS Thesis students finish under his supervision. His research interests are in parallel computation, simulation, computer graphics, and virtual reality. He is also a Senior Member of the ACM, and a Senior Member of the International Society for Computers and their Applications (ISCA).

Non-Parametric Error Estimation for σ-AQP using Optimized Bootstrap Sampling

Feng Yu[*†]
Youngstown State University, Youngstown, OH 44555, USA
Semih Cal[‡]
Texas Tech University, Lubbock, TX 79409, USA
En Cheng[§]
University of Akron, Akron, OH 44325, USA
Lucy Kerns[¶]
Youngstown State University, Youngstown, OH 44555, USA
Weidong Xiong[‖]
Cleveland State University, Cleveland, OH 44115, USA

Abstract

Approximate query processing (or AQP) aims to quickly provide approximated answers for time-consuming search queries on large datasets. It brings enormous benefits in data science when the query execution efficiency weighs more than the accuracy. However, assessing the accuracy of an approximated answer from AQP still lacks study. Existing work usually relies on strict dataset assumptions that are often not satisfied in real-world datasets. In this work, we employ a non-parametric statistical method, called bootstrap sampling, to assess errors of an AQP system for selection queries (or σ-AQP). We implement a prototype AQP system integrated with a bootstrap sampling engine that can estimate the standard deviation and produce confidence intervals for selection query estimations. Extensive experiments operating the prototype system demonstrated that the confidence intervals generated can cover the ground truth query results with high accuracy and low computing costs. In addition, we introduce optimization strategies for bootstrap sampling which can improve the overall computing efficiency of the prototype AQP system.

Key Words: Approximate query processing, error estimation, non-parametric method, bootstrap sampling

1 Introduction

Efficient query processing of complex queries on big data posts a demanding challenge for modern data management systems. Much work has been developed towards promptly executing data queries on both hardware and software platforms [9, 21, 22]. However, calculating the exact answer for each data query is expensive and may not be necessary for all scenarios. For example, during the exploratory data analysis (or EDA), a user often only needs approximated answers for a collection of testing queries where the execution speed weighs more than the accuracy.

Approximate query processing (or AQP) is an alternative scheme to provide estimated query answers with satisfying accuracy and within a short time [2, 18, 19]. AQP doesn't need to run the query on the original dataset but can collect statistics to generate query estimations. A common application of AQP is to estimate selection (or σ) queries, called σ-AQP. For selection queries, simple random samples are usually employed for fast and accurate query estimations [1, 24].

An open question for the AQP research is how to efficiently assess the error of query estimation. The challenge is that, for different selection conditions, the underlying distributions of the result sets are different and difficult to predict. This creates an obstacle to efficiently assessing the estimation errors for AQP systems.

Bootstrap sampling [23] is a statistical technique that can assess the errors of sample-based estimators. One advantage of bootstrap sampling is that it doesn't rely on any knowledge of the data distribution to provide error estimation, but can "pull itself up by its bootstrap". It conducts a special sampling method, called resampling, which generates many replicated random samples with replacement, called bootstrap samples, from the original random samples used by AQP. Using the bootstrap samples, common error assessments, such as the standard deviation, of a query estimator can be calculated. An advantage of bootstrap sampling is it doesn't require restricted assumptions of data such as normal distribution used by large number theory. Statistical methods like these are commonly named as non-parametric methods [12].

In this work, we will focus on using bootstrap sampling to assess the estimation error of a σ-AQP system. The contributions of this work are as follows:

1. We propose a framework equipped with the bootstrap sampling method to assess the errors of an AQP system

[*] Email addresses are to be put first. fyu@ysu.edu, scal@ttu.edu, echeng@uakron.edu, xlu@ysu.edu, w.xiong15@csuohio.edu

[†] Department of Computer Science and Information Systems.

[‡] Department of Computer Science.

[§] Department of Computer Science.

[¶] Department of Mathematics and Statistics.

[‖] Department of Electrical Engineering and Computer Science.

for selection queries (or σ-AQP). A prototype system is implemented to simulate a real-world database system that can execute common selection queries. This system is integrated with a bootstrap sampling engine used for non-parametric error assessment.

2. We test the performance of the prototype system on multiple datasets with various combinations of hyper-parameters to simulate real-world scenarios. The experimental results show satisfying accuracy of error assessment.
3. With the findings of the computing bottlenecks of the bootstrap sampling procedure, we propose optimization schemes to improve the overall system performance.

Compared with the conference version work [4], additional contributions are made including:

1. We extended the sections of background and bootstrap sampling framework. We added the related work of AQP and bootstrap sampling.
2. We performed extended experiments of error assessment. Various datasets with skewness were employed in the accuracy tests of bootstrap confidence intervals.
3. Additional analysis of the error assessment experiments is included. The means and standard deviations of the confidence interval hit ratios and the bootstrap standard deviations are presented.

The rest of this work is organized as follows. Section 2 introduces the background of σ-AQP and bootstrap sampling. Section 3 describes how to use bootstrap sampling to assess estimation errors from a sample-based σ-AQP scheme. The implementation of the prototype system incorporating σ-AQP and a bootstrap sampling engine is described in Section 4. Experimental results are presented in Section 5. The related work is included in Section 6. The conclusion and future work are included in Section 7.

2 Background

2.1 σ-AQP

Approximate query processing (or AQP) is the technology to provide approximated answers to complex queries using statistical methods. It aims to provide accurate query estimations within a short time frame. σ-AQP is the AQP focusing on estimating selection (SELECT or σ) queries. Given a selection query Q on a table R, to get the ground truth query answer Y_{GT}, the traditional scheme of query answering is to execute query Q on R which may take a long time when R has a large volume and the query result size is big. Instead of running query Q directly on the original dataset R, σ-AQP takes a simple random sample without replacement (SRSOR) from R, denoted by S, and runs the query Q on S to get a sample result Y_s. In this case, the ground truth, Y_{GT} can then be estimated by

$$Y_{GT} = \frac{Y_s}{f} \tag{1}$$

where $f = |S|/|R|$ is the sampling ratio.

Figure 1 demonstrates an example of simple random sampling without replacement. If the original table includes 100 records, using a 20% sampling ratio, 20 records will be randomly selected without replacement and saved into a sample table. After that, σ-AQP will use the sample table to produce estimations for selection queries with a sampling ratio parameter set to 20%. In practice, the sampling ratio is usually tiny. For example, a sampling ratio of less than 1% is usually employed. For highly skewed data, a larger sampling ratio can be used to increase the accuracy of query estimation.

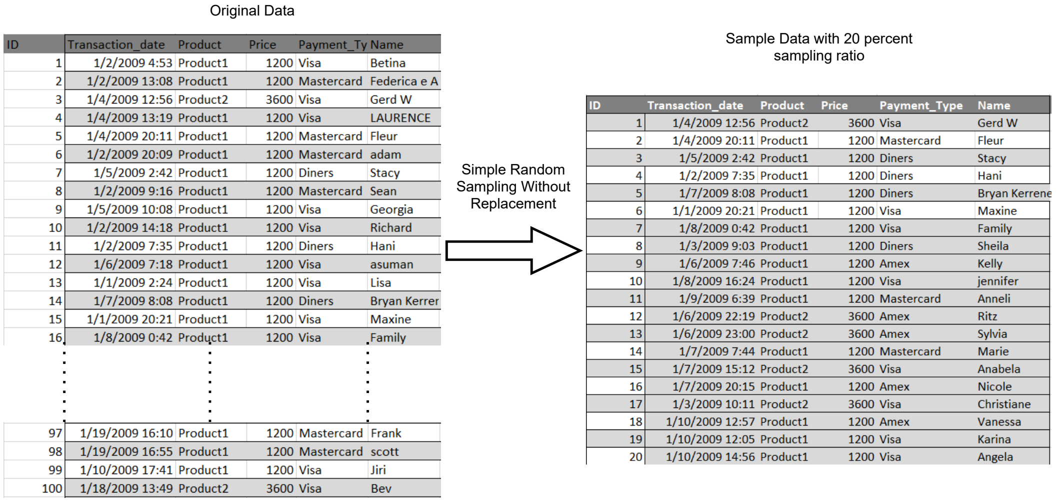
2.2 Bootstrap Sampling

Bootstrap sampling was originally introduced by Bradley Efron in 1979 [10]. It is a computer-assisted method designed to measure the quality of various statistical estimators. Bootstrap sampling generates a collection of new distributions from the original distribution and can derive their variance which can be used to quantify the accuracy of statistical estimators based on the observed data. It works well when the target data is drawn from unknown distributions, which is superior to deriving closed-form methods based on limited data assumptions.

A unique statistical feature in bootstrap sampling is *resampling*. This procedure generates new distributions, called *bootstrap samples*, from a given sampled dataset using simple random sampling with replacement (SRSWR). Each resampled new distribution can produce a scalar called a *bootstrap replication*. Bootstrap sampling generates a large number of bootstrap replications and can use them to estimate the statistical features, such as standard deviation, of the originally given dataset even when the original distribution is unknown.

Figure 2 depicts a simple example of how bootstrap sampling is performed. When given a sample data $\mathbf{y} = (y_1, y_2, ..., y_n)$ from an unknown distribution F, a *bootstrap sample* $\mathbf{y}^* = (y_1^*, y_2^*, ..., y_n^*)$ is a resampled collection obtained by randomly sample n times with replacement from the original sample y_1, y_2, ..., y_n. For instance, if $n = 5$, we might obtain different bootstrap samples, such as $\mathbf{y}_1^* = (y_5, y_3, y_1, y_2, y_1)$, $\mathbf{y}_2^* = (y_2, y_5, y_4, y_1, y_2)$, $\mathbf{y}_3^* = (y_3, y_3, y_2, y_3, y_4)$, etc. These resamples are shown in Figure 2a. Figure 2b depicts a bootstrap sample example.

A useful application of bootstrap sampling is to estimate the standard deviation of a statistical estimator from an unknown distribution. Suppose we wish to estimate an unknown population parameter, $\theta = t(F)$, based on the sampled data $\mathbf{y}$, i.e., $\hat{\theta} = s(\mathbf{y})$. We first generate a number of B independent bootstrap samples. Given each bootstrap sample $\mathbf{y}^*$, a *bootstrap replication* of $\hat{\theta}$ is computed as $\hat{\theta}^* = s(\mathbf{y}^*)$. For example, if $\hat{\theta}$ is the sample mean $\bar{\mathbf{y}}$, a bootstrap replication $\hat{\theta}^*$ is the mean of a bootstrap sample $\overline{\mathbf{y}^*}$. The standard error of $\hat{\theta}^*$, i.e. $\widehat{se}_B(\hat{\theta}^*)$, called the *bootstrap estimation of standard error*, can be calculated from the B bootstrap replications as follows.



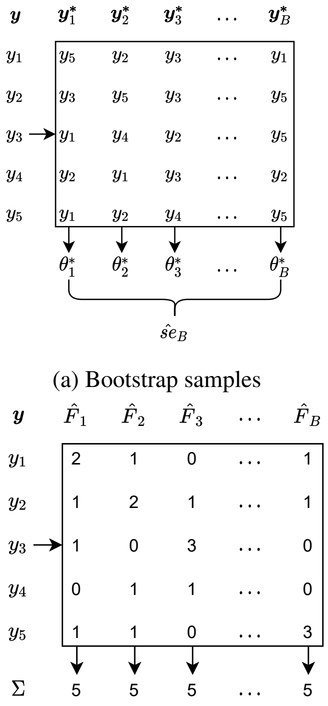
Original Data

ID	Transaction_date	Product	Price	Payment_Ty	Name
1	1/2/2009 4:53	Product1	1200	Visa	Betina
2	1/2/2009 13:08	Product1	1200	Mastercard	Federica e A
3	1/4/2009 12:56	Product2	3600	Visa	Gerd W
4	1/4/2009 13:19	Product1	1200	Visa	LAURENCE
5	1/4/2009 20:11	Product1	1200	Mastercard	Fleur
6	1/2/2009 20:09	Product1	1200	Mastercard	adam
7	1/5/2009 2:42	Product1	1200	Diners	Stacy
8	1/2/2009 9:16	Product1	1200	Mastercard	Sean
9	1/5/2009 10:08	Product1	1200	Visa	Georgia
10	1/2/2009 14:18	Product1	1200	Visa	Richard
11	1/2/2009 7:35	Product1	1200	Diners	Hani
12	1/6/2009 7:18	Product1	1200	Visa	asuman
13	1/1/2009 2:24	Product1	1200	Visa	Lisa
14	1/7/2009 8:08	Product1	1200	Diners	Bryan Kerrer
15	1/1/2009 20:21	Product1	1200	Visa	Maxine
16	1/8/2009 0:42	Product1	1200	Visa	Family
	⋮	⋮			⋮
97	1/19/2009 16:10	Product1	1200	Mastercard	Frank
98	1/19/2009 16:55	Product1	1200	Mastercard	scott
99	1/10/2009 17:41	Product1	1200	Visa	Jiri
100	1/18/2009 13:49	Product2	3600	Visa	Bev

Sample Data with 20 percent sampling ratio

ID	Transaction_date	Product	Price	Payment_Type	Name
1	1/4/2009 12:56	Product2	3600	Visa	Gerd W
2	1/4/2009 20:11	Product1	1200	Mastercard	Fleur
3	1/5/2009 2:42	Product1	1200	Diners	Stacy
4	1/2/2009 7:35	Product1	1200	Diners	Hani
5	1/7/2009 8:08	Product1	1200	Diners	Bryan Kerrene
6	1/1/2009 20:21	Product1	1200	Visa	Maxine
7	1/8/2009 0:42	Product1	1200	Visa	Family
8	1/3/2009 9:03	Product1	1200	Diners	Sheila
9	1/6/2009 7:46	Product1	1200	Amex	Kelly
10	1/8/2009 16:24	Product1	1200	Visa	jennifer
11	1/9/2009 6:39	Product1	1200	Mastercard	Anneli
12	1/6/2009 22:19	Product2	3600	Amex	Ritz
13	1/6/2009 23:00	Product2	3600	Amex	Sylvia
14	1/7/2009 7:44	Product1	1200	Mastercard	Marie
15	1/7/2009 15:12	Product2	3600	Visa	Anabela
16	1/7/2009 20:15	Product1	1200	Amex	Nicole
17	1/3/2009 10:11	Product2	3600	Visa	Christiane
18	1/10/2009 12:57	Product1	1200	Amex	Vanessa
19	1/10/2009 12:05	Product1	1200	Visa	Karina
20	1/10/2009 14:56	Product1	1200	Visa	Angela

Figure 1: Example: simple random sampling without replacement (SRSWOR) using 20% sampling ratio



$\boldsymbol{y}$	$\boldsymbol{y}_1^*$	$\boldsymbol{y}_2^*$	$\boldsymbol{y}_3^*$	$\ldots$	$\boldsymbol{y}_B^*$
y_1	y_5	y_2	y_3	$\ldots$	y_1
y_2	y_3	y_5	y_3	$\ldots$	y_5
y_3	y_1	y_4	y_2	$\ldots$	y_5
y_4	y_2	y_1	y_3	$\ldots$	y_2
y_5	y_1	y_2	y_4	$\ldots$	y_5
	θ_1^*	θ_2^*	θ_3^*	$\ldots$	θ_B^*

$$\underbrace{\theta_1^* \quad \theta_2^* \quad \theta_3^* \quad \ldots \quad \theta_B^*}_{\widehat{se}_B}$$

(a) Bootstrap samples

$\boldsymbol{y}$	$\hat{F}_1$	$\hat{F}_2$	$\hat{F}_3$	$\ldots$	$\hat{F}_B$
y_1	2	1	0	...	1
y_2	1	2	1	...	1
y_3	1	0	3	...	0
y_4	0	1	1	...	0
y_5	1	1	0	...	3
Σ	5	5	5	...	5

(b) Bootstrap distributions

Figure 2: Example: bootstrap sampling

$$\widehat{se}_B(\hat{\theta}^*) = \left[\frac{1}{B-1}\sum_{i=1}^{B}\left(\hat{\theta}_i^* - \bar{\theta}^*\right)^2\right]^{\frac{1}{2}} \tag{2}$$

where $\bar{\theta}^* = \sum_{i=1}^{B}\hat{\theta}_i^*/B$.

When $B \to \infty$, we have $\widehat{se}_B(\hat{\theta}^*) \to se_{\hat{F}}(\hat{\theta}^*)$, where $se_{\hat{F}}(\hat{\theta}^*)$ is called the *ideal bootstrap estimation* of the ground truth standard error of $\hat{\theta}$, i.e. $se_F(\hat{\theta})$. Both $se_{\hat{F}}(\hat{\theta}^*)$ and its approximation $\widehat{se}_B(\hat{\theta}^*)$ are called *non-parametric bootstrap* estimates since they are generated from the distributions, $\hat{F}$, which are non-parametric estimates of the ground truth population F.

3 Bootstrap for Selection Query Error Estimation

3.1 Selection Query Estimation

We consider the following query formulation in this research:

```
Q: SELECT Aggregation(attribute collection)
FROM table_name WHERE conditions
```

After a query Q is executed on the sample table S, each sample tuple $u_i \in S$ will produce a tuple query result y_i based on the aggregation function. For example, if the aggregation function is COUNT, then y_i is either 1 if u_i satisfies the selection condition or 0 otherwise. The query result Y_s on the sample table S is calculated as $Y_s = \sum_{i=1}^{n} y_i$, where $n = |S|$ is the sample size. Suppose the size of the original table R is N, and the sample fraction $f = \frac{n}{N}$, then the estimation of the query result ground truth is

$$\widehat{Y} = \frac{Y_s}{f} \tag{3}$$

This estimation works well if the original table R has low skewness and the sample S is uniformly collected from R. Otherwise, the accuracy may be low when data is highly skewed or the sample S is not uniformly distributed (or even includes correlation).

3.2 Bootstrap Sampling from Query Results

After the sample query results $S_Q = \{y_i\}_{i=1}^{n}$ are obtained by executing Q on the sample relation S, bootstrap samples $\{\mathbf{y}_j^*\}_{j=1}^{B}$ can be generated for error estimation, where B is the total times of bootstrap sampling. Each $\mathbf{y}_j^* = \{y_{j,i}^*\}_{i=1}^{n}$ a bootstrap sampling of S_Q, where each query result $y^*_{j,i}$, $1 \le i \le n$, is randomly sampled with replacement from S_Q.

To obtain the bootstrap replication, we use the same estimator in Eq (2) on each $\mathbf{y}^*_j$, $j = 1, ..., B$ as

$$\widehat{Y}^*_j = \frac{Y_{\mathbf{y}^*_j}}{f} \tag{4}$$

For example, if the aggregation is COUNT, then the estimator is

$$\widehat{Y}^*_j = \frac{1}{f}\sum_{i=1}^{n} y^*_{j,i} \tag{5}$$

After repeating the bootstrap sampling for B times, a collection of bootstrap replications is obtained, denoted by $\widehat{Y}^*_B = \{\widehat{Y}^*_j\}^B_{j=1}$. The bootstrap standard deviation is calculated as

$$\widehat{se}_B = \left[\frac{\sum_{j=1}^{B}(\widehat{Y}^*_j - \overline{\widehat{Y}^*_B})^2}{B-1}\right]^{\frac{1}{2}} \tag{6}$$

where $\overline{\widehat{Y}^*_B}$ is the sample mean of all bootstrap replications $\widehat{Y}^*_B$. By the theory of bootstrap sampling, we claim that Eq (6) is the *bootstrap estimation of the standard error* of $\widehat{Y}$ which estimates the query result Y_{GT} of query Q.

3.3 Computing the Confidence Interval

There are different methods in bootstrap sampling to generate a confidence interval (or CI), such as the normal-theory CI, bootstrap percentile CI, and basic bootstrap CI. There are also improved CI methods to increase the accuracy such as the Bias-Corrected and Accelerated interval (BC$_a$) and Approximate Bootstrap Confidence interval (ABC) [10]. In this work, we implemented the normal-theory CI method, which is calculated as

$$\left(\widehat{Y} - z_{\alpha/2}\cdot\widehat{se}_B, \widehat{Y} + z_{\alpha/2}\cdot\widehat{se}_B\right) \tag{7}$$

where $\widehat{Y}$ is the query estimation from AQP, $1-\alpha \in [0,1]$ is the confidence level, and $z_{\alpha/2}$ is the upper-$\alpha/2$ standard normal critical point. For example, for a 90% confidence level (i.e., $\alpha = .10$), $z_{.05} = 1.645$, and for a 95% confidence level, $z_{.025}$=1.960.

4 Implementation

We propose a prototype AQP system with the ability to generate error estimations using bootstrap sampling. The prototype system consists of the following parts: a simple query processor, a query execution engine for selection, a σ-AQP engine using simple random sampling, and a bootstrap engine for error estimation. The architecture of the query processor is depicted in Figure 3.

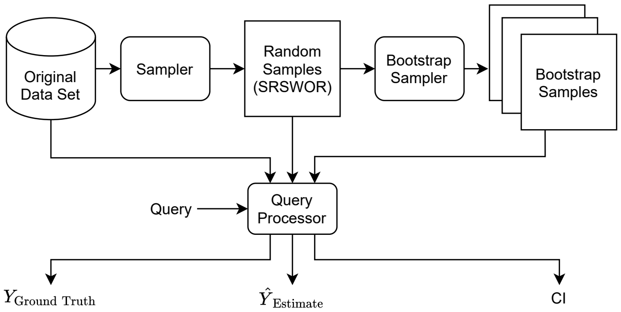


Figure 3: Prototype σ-AQP framework with a bootstrap sampling engine

The implemented query processor reads queries from a plain text file and executes them accordingly. The query execution engine reads each tuple from the table data file and produces a tuple query result by checking whether it satisfies the selection condition. Summarizing all tuple query results will produce the final query result.

The σ-AQP engine of this system has two functions: generates a sample table S from the base table R using simple random sampling without replacement (simple random sampler) and provide query estimations using the sample table (sample estimator). When sampling starts, a series of random row numbers will be generated in an array and the sampler will access the base table file and retrieve only the tuples in the random number array. Depending on the volume of sample tuples, if the sample tuples cannot fit into the memory, the sampler will output the sampled tuples into the sample table file in batches. Otherwise, the sampled tuples will be read in one batch and saved into the sample table.

After the sample tuples are drawn, the sample estimator of the σ-AQP engine can produce a query estimation by first executing the original query Q on the sample table S and getting a sample result set Y_s. The query execution engine will be called to run the query on the sample table S and the sample query results S_Q will be generated. The estimation of the query result, $\widehat{Y}_{est}$, will be calculated using Eq (3). The bootstrap engine will perform bootstrap sampling on the sample query results S_Q, calculate the bootstrap standard deviation $\widehat{se}_B$, and produce the bootstrap confidence interval (CI) using Eq (7).

5 Experiment

We present the experimental results in this section. First, we test the error assessment accuracy of the implemented prototype AQP system. Second, we investigate the performance of the bootstrap sampling procedure during the accuracy tests. Finally, we present the performance results using optimized bootstrap sampling methods.

5.1 Experiment Setup

The experiment server is equipped with an Intel Xeon E5-1620 v4 CPU and 8GB of RAM and runs CentOS 7 Linux. The experiment code is written in C and Python languages. The major prototype components such as query parsing, query processing, AQP, and bootstrap sampling module

Table 1: Test queries for accuracy experiments

No.	Query
1	select count(*) from lineitem where L_QUANTITY <20 and L_QUANTITY >0
2	select count(*) from lineitem where L_LINENUMBER <3 and L_LINENUMBER >0
3	select count(*) from lineitem where L_LINENUMBER <5 and L_LINENUMBER >2
4	select count(*) from lineitem where L_DISCOUNT <.07 and L_DISCOUNT >.02
5	select count(*) from lineitem where L_EXTENDEDPRICE <100000.00 and L_EXTENDEDPRICE >20000.00
6	select count(*) from lineitem where L_DISCOUNT <.04 and L_DISCOUNT >0.0
7	select count(*) from lineitem where L_QUANTITY <20 and L_QUANTITY >10
8	select count(*) from lineitem where L_DISCOUNT <.05 and L_DISCOUNT >.02
9	select count(*) from lineitem where L_EXTENDEDPRICE <15000.00 and L_EXTENDEDPRICE >0.0
10	select count(*) from lineitem where L_LINENUMBER <2 and L_LINENUMBER >0

are implemented in C. The driver programs for experiments are written in Python. The source code of the prototype system and experiments are available on GitHub[1].

The tests datasets are generated using the TPC-H benchmark with skew[2] [8] which is widely used for data querying tests. We focused on testing SELECT (or σ) queries for σ-AQP error assessment and we chose the largest table, namely LINEITEM, in a TPC-H database. We generated multiple TPC-H datasets (only including the LINEITEM table) in volumes of 100MB, 1GB, and 10GB and with skewness of 0 (no skew) and 1 (highly skewed), respectively. We randomly generated 10 test queries with different selection ranges listed in Table 1. Among them, five queries are large-range selection queries and five queries are small-range selection queries.

5.2 Bootstrap Accuracy Tests

We estimate each test query using the implemented AQP system which produces a 95% level bootstrap CI as a range estimation. The implemented query processor computes the ground truth of the query, i.e. Y_{GT}, on the original dataset. If the Y_{GT} is contained in the CI, it's considered a "hit"; otherwise, it's a "miss". For each test query, we repeatedly generate the bootstrap CI for 10 times and calculate the averaged hit ratio as follows.

$$\text{hit ratio} = \frac{\text{count(CI includes } Y_{GT})}{\text{count(overall experiments)}} \times 100\% \qquad (8)$$

Figure 4 depicts the results of accuracy tests using bootstrap sampling. Each small figure depicts the result on one test dataset using different sampling ratios (f) including 0.1%, 0.5%, and 1%. To study how the bootstrap iterations (B) affect the hit ratios, we use compare tests with B=200 and B=2000 (recommended in [10]). In addition, to study how the data skew (z) affects the hit ratios, we compute hit ratios with different skewness values, z=0 (no skew) and z=1 (highly skewed).

Table 2 includes the averaged hit ratios of all experiments. The overall averaged hit ratios range between 94.8% to 97.6%. As observed in the results, a higher sampling ratio usually generates higher hit ratios. However, comparing the hit ratio results with B=200 and those with B=2000, no significant differences in hit ratios were observed. For instance, the overall averaged hit ratios of z=1 when B=200 (97.9%) and B=2000 (97.6%) are both slightly higher than those with z=0 when B=200 (95.8%) and B=2000 (94.8%).

Table 3 includes the standard deviations of the hit ratios. The overall standard deviations range between 4.8 to 6.1. First, for each fixed value of B and z, the standard deviation of hit ratios generally decreases while the sampling ratio increases. For example, when B=2000 and z=1, the standard deviations for sampling ratios 0.1%, 0.5%, and 1% are 9.5, 6.7, 6.3, respectively. On the other hand, when B and z changed, no significant difference in the standard deviations were observed. The overall standard deviations of more skewed data when z=1 (STD=4.8 for B=200 and 5.6 for B=2000) are both slightly smaller than those when z=0 (STD=6.1 for B=200 and 6.4 for B=2000).

Table 4 includes the bootstrap standard deviations. A smaller bootstrap standard deviation means a smaller error estimation of the query estimation and a narrower bootstrap CI. As observed, the bootstrap standard deviations generally decrease with the sampling ratio increases. This demonstrates that query estimations are more consistent given higher sampling ratios. Changes of B and z do not significantly affect the bootstrap standard deviations.

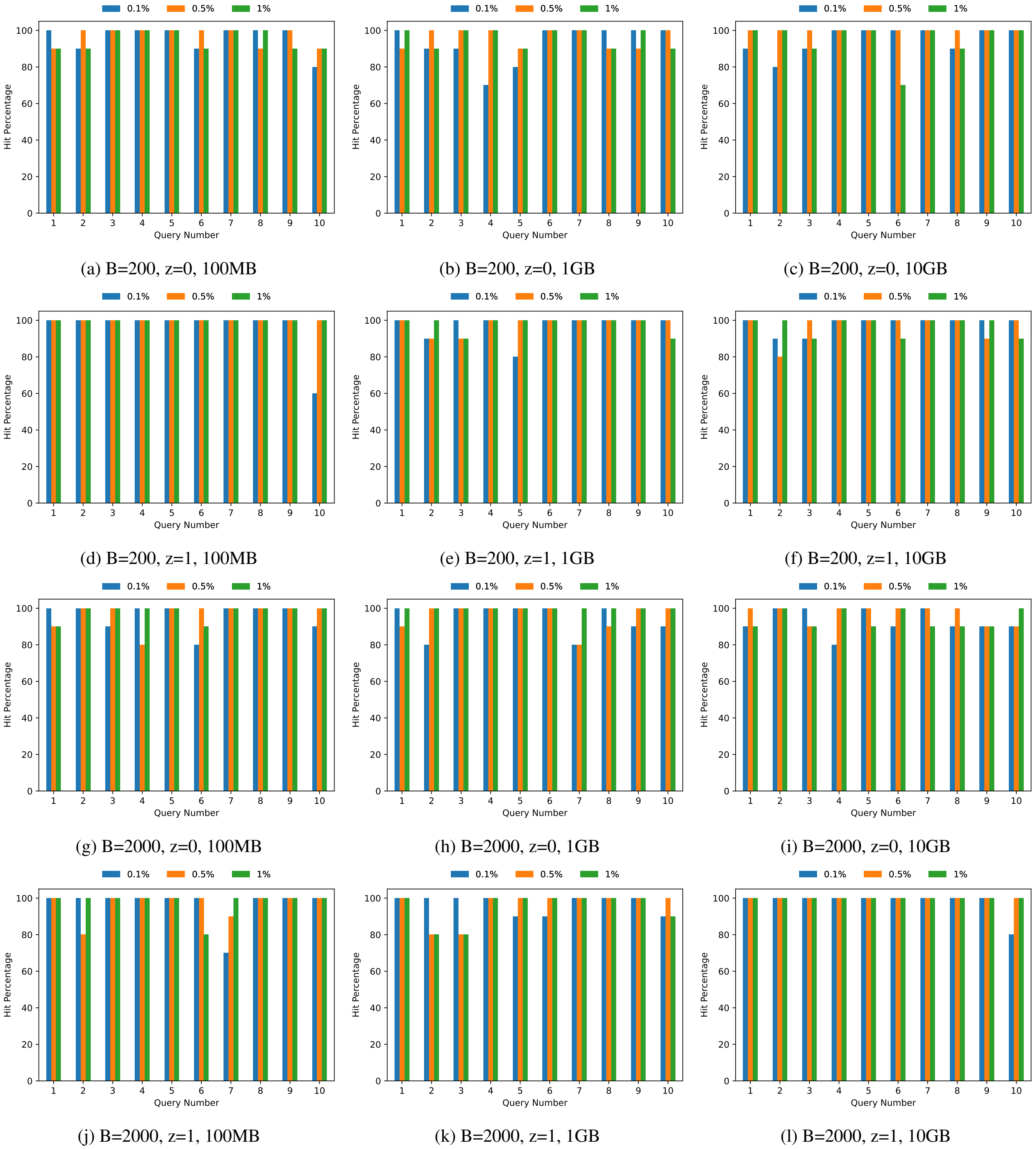
In general, the experiments show that higher sampling ratios help to improve bootstrap CI hit ratios. On the other hand, the changes of bootstrap iterations (B) or data skewness (z) do not significantly impact the error assessment accuracy.

5.3 Speed Performance Tests

We present the speed performance results when estimating the test queries on the 1GB dataset in Figure 5. The running time to answer each test query is composed of three parts including the file access time, simple random sampling time, and bootstrap sampling time. The file accessing, random sampling, and

[1] The experiment code is available at `https://github.com/YSU-Data-Lab/Semih_Cal_Thesis_Summer_2021`

[2] We employed the TPC-H toolkit available at `https://github.com/YSU-Data-Lab/TPC-H-Skew`



(a) B=200, z=0, 100MB

(b) B=200, z=0, 1GB

(c) B=200, z=0, 10GB

(d) B=200, z=1, 100MB

(e) B=200, z=1, 1GB

(f) B=200, z=1, 10GB

(g) B=2000, z=0, 100MB

(h) B=2000, z=0, 1GB

(i) B=2000, z=0, 10GB

(j) B=2000, z=1, 100MB

(k) B=2000, z=1, 1GB

(l) B=2000, z=1, 10GB

Figure 4: Hit ratios of 95% level bootstrap confidence intervals (B: bootstrap iterations; sampling ratio: 0.1%, 0.5%, and 1%; z - data skewness: 0 or 1, larger value is more skewed)

bootstrap sampling procedures did not use any memory buffer to simulate the worst performance scenario. The groups of the first three figures and the last three figures show that, when the total bootstrap iterations (B) stay the same and the sampling ratio (f) increases, the bootstrap sampling time increases and becomes a major bottleneck compared with the random sampling time. The same trend is also observed when f stays the same and B increases, for example, comparing Figure 5a and Figure 5d. Therefore, the performance of bootstrap sampling is mainly affected by values of B and f, especially when the data resides out-of-core.

5.4 Tests of Optimized Bootstrap Sampling

To increase the overall performance the prototype AQP system, we improve the bootstrap sampling procedure in the following aspects.

1. To lower the data access time, the tuples for bootstrap sampling are not directly accessed. It's only the query sample results that are calculated and stored in a memory array which are passed to the bootstrap engine. Since the sampling ratios for AQP are usually small (less than 1%), these arrays shall be small enough to fit into the main memory. If the sampled array is too large, other alternatives can be implemented such as using partitioned data arrays.
2. The sorting of the random numbers for resampling is omitted to save computation. During the resampling procedure, the bootstrap random numbers are kept unsorted. After that, a resample array of sample query results are extracted according to the bootstrap random number array by in-memory array mapping. For example, if the generated random number array for resampling is $\{5,3,1,2\}$, then the query results resampled shall be $\{y_5,y_3,y_1,y_2\}$.

We perform the same experiments on the 1GB test data using the prototype system with the optimized bootstrap sampling engine. Figure 6 depicts the execution time speedup factors comparing the optimized bootstrap sampling scheme with the original scheme. The speedup factor is defined as follows.

$$\text{speedup factor} = \frac{\text{time(original bootstrap sampling)}}{\text{time(optimized bootstrap sampling)}} \quad (9)$$

As observed from the experiment results, the optimized system reached an averaged speed-up factor of 5 comparing the bootstrap sampling execution times. It also reached an averaged speedup factor of 2 comparing the file access times. In addition, the speedup factor progressively increases with the sampling ratio.

6 Related Work

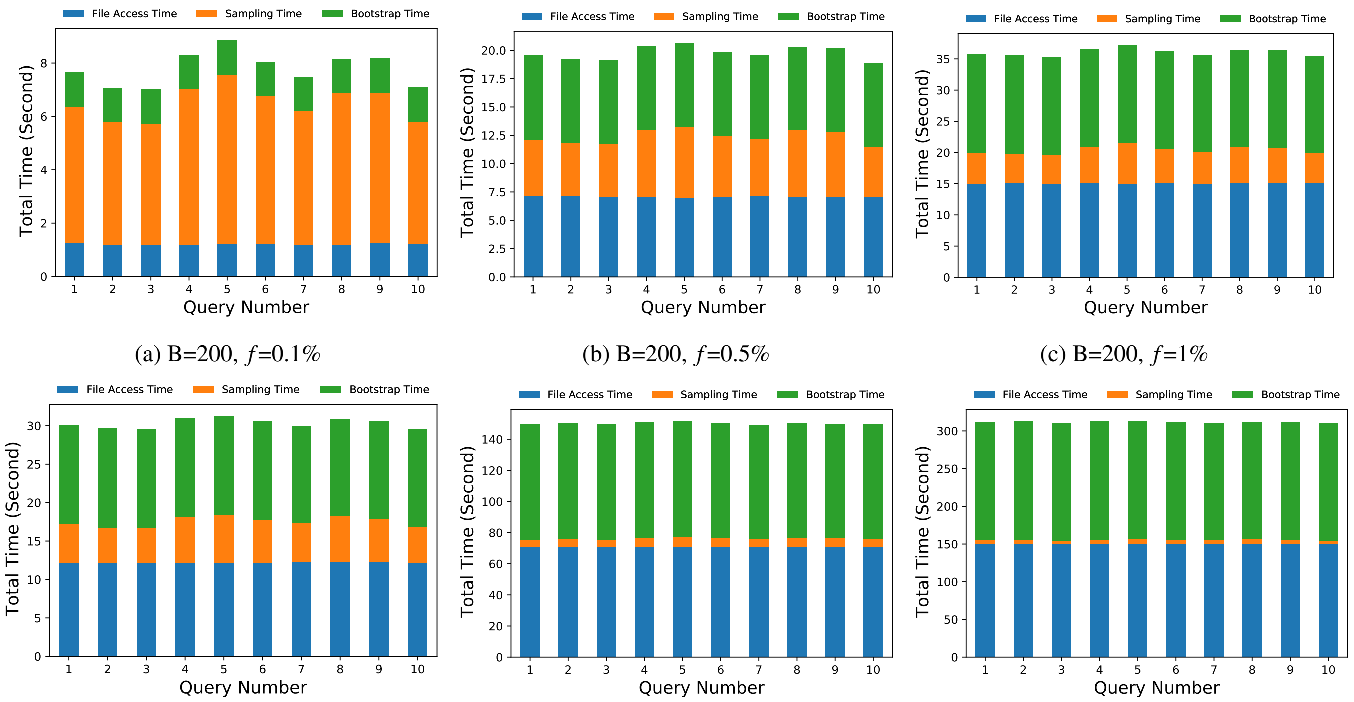
Based on the schemes of statistics collection, AQP can be categorized into two directions including the *online AQP* and the *offline AQP* [5]. The online AQP schemes [6, 16, 17], by the name, start collecting statistics only after the target query for approximation is given. Therefore, to reach a high statistics collecting speed, they usually rely on auxiliary data structures, such as indices and hash tables. Creating and maintaining these auxiliary data structures will generate heavy overheads especially for big data applications. Another drawback is that their collected statistics can only be used once for a given query, and must be re-collected for a different query which wastes computing resources. The offline AQP [1, 24], on the other hand, collects statistics before a query is submitted. It usually needs the knowledge of the whole database schema or the join graph, to create a holistic statistical synopsis. One advantage of the offline AQP is it doesn't rely on any auxiliary data structure or advanced hardware to collect statistics because the statistics collection happens before the target query is given and doesn't affect the run-time system performance. Another advantage is the statistics collected by offline AQP schemes are reusable for all future target queries given the database join graph is not changed.

Bootstrap sampling has a long history in statistics. [12] is a definitive book for its literature. However, this powerful method still waits to be fully utilized in modern database systems. Existing work [10, 11, 20, 7, 3, 15, 14, 13, 25, 26, 27] has made contribution to this direction. Among them, Pol and Jermaine in [20] focused on increasing the performance of bootstrap sampling by lowering the number of bootstrap iterations. To this end, a new data structure named resampling tree was introduced in their proposed ODM framework. Zeng et al. [25, 26] introduced an improved method called Analytical Bootstrap Method (ABM) that can avoid bootstrap iterations for limited types of database queries. Kleiner et al. [14] introduced a new bootstrap sampling method that can reduce bootstrap iterations on big datasets.

Our work concentrates on the empirical analysis of the bootstrap sampling for σ-AQP systems. We claim that the mentioned related work doesn't fully address the topics in this work. However, some methods [20, 25, 14] can help to improve the bootstrap sampling performance in this work.

7 Conclusion and Future Work

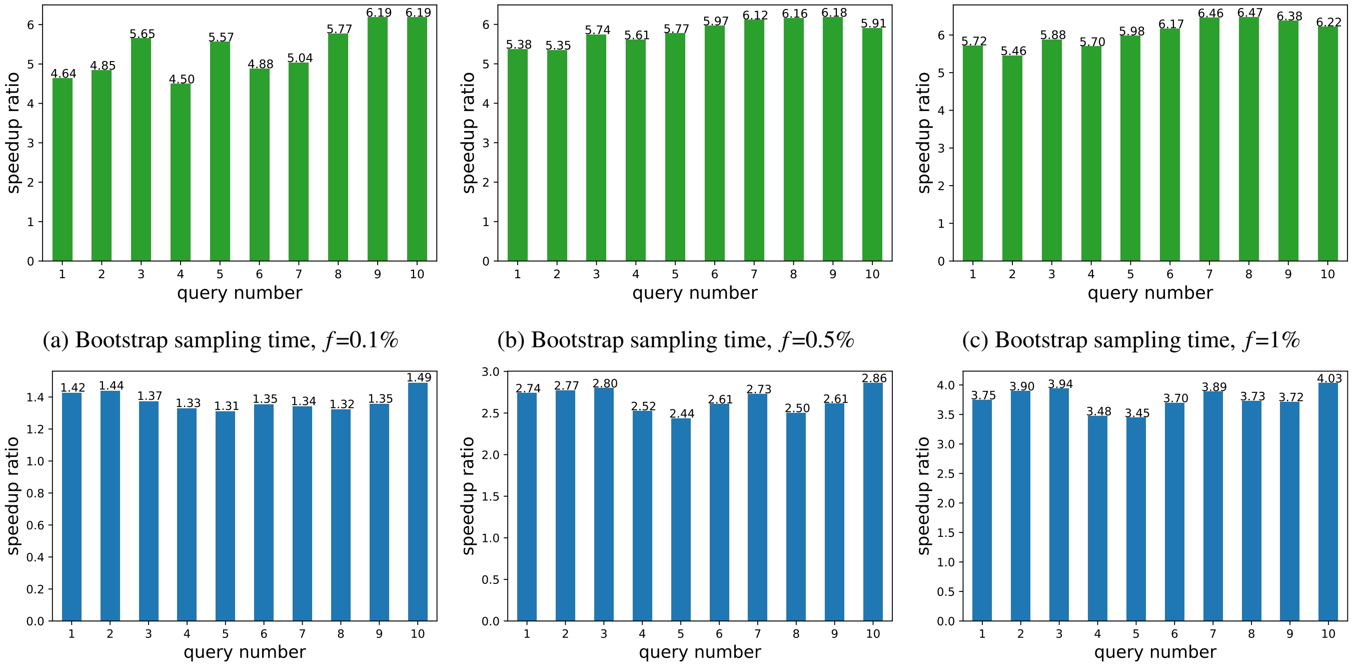
In this work, we employ a non-parametric statistical method, called bootstrap sampling, to assess the estimation errors of σ-AQP systems. The contributions are threefold. First, we developed a prototype σ-AQP system integrated with a bootstrap sampling engine that can produce confidence intervals for selection query estimations. Second, we performed extensive query estimation experiments using the implemented system. The results showed that the bootstrap confidence intervals produced are highly accurate even when small sampling ratios were used. Third, we studied the performance bottlenecks of the implemented system and proposed multiple strategies to optimize the bootstrap sampling procedure which were shown effective in experiments. In the future, we will



(a) B=200, *f*=0.1%
(b) B=200, *f*=0.5%
(c) B=200, *f*=1%

(d) B=2000, *f*=0.1%
(e) B=2000, *f*=0.5%
(f) B=2000, *f*=1%

Figure 5: Time overheads in 1GB data tests (B: bootstrap iterations, *f*: sampling ratio (%))



(a) Bootstrap sampling time, *f*=0.1%
(b) Bootstrap sampling time, *f*=0.5%
(c) Bootstrap sampling time, *f*=1%

(d) Total time, *f*=0.1%
(e) Total time, *f*=0.5%
(f) Total time, *f*=1%

Figure 6: Speedup ratios using optimized bootstrap sampling in 1GB data tests (*f*: sampling ratio (%))

Table 2: Hit ratio means (%, B: total bootstrap iterations, z: skewness value)

B	z	100 MB				1GB				10GB				Overall
		0.1%	0.5%	1%	avg	0.1%	0.5%	1%	avg	0.1%	0.5%	1%	avg	avg
200	0	96.0	97.0	95.0	96.0	93.0	96.0	96.0	95.0	95.0	99.0	95.0	96.3	95.8
200	1	96.0	100.0	100.0	98.7	97.0	98.0	98.0	97.7	98.0	97.0	97.0	97.3	97.9
2000	0	93.0	88.0	99.0	93.3	94.0	95.0	97.0	95.3	97.0	97.0	93.0	95.7	94.8
2000	1	97.0	97.0	98.0	97.3	97.0	96.0	95.0	96.0	98.0	100.0	100.0	99.3	97.6

Table 3: Hit ratio standard deviations (B: total bootstrap iterations, z: skewness value)

B	z	100 MB				1GB				10GB				Overall
		0.1%	0.5%	1%	avg	0.1%	0.5%	1%	avg	0.1%	0.5%	1%	avg	avg
200	0	7.0	4.8	5.3	5.7	10.6	5.2	5.2	7.0	7.1	0.0	9.7	5.6	6.1
200	1	12.6	0.0	0.0	4.2	6.7	4.2	4.2	5.0	4.2	6.7	4.8	5.2	4.8
2000	0	9.5	11.4	3.2	8.0	5.2	7.1	4.8	5.7	6.7	4.8	4.8	5.4	6.4
2000	1	9.5	6.7	6.3	7.5	4.8	8.4	8.5	7.2	6.3	0.0	0.0	2.1	5.6

Table 4: Bootstrap standard deviations (B: total bootstrap iterations, z: skewness value)

B	z	100 MB				1GB				10GB				Overall
		0.1%	0.5%	1%	avg	0.1%	0.5%	1%	avg	0.1%	0.5%	1%	avg	avg
200	0	10817.8	4844.3	3449.9	6370.7	34567.3	15368.4	10775.8	20237.2	108118.6	48151.8	34063.4	63444.6	30017.5
200	1	10821.6	4889.6	3477.0	6396.1	34044.4	15211.9	10851.6	20036.0	107887.7	47909.3	34450.6	63415.9	29949.3
2000	0	10894.7	4855.3	3427.2	6392.4	34198.9	15371.4	10823.5	20131.3	108599.9	48490.2	34193.6	63761.2	30095.0
2000	1	10864.4	4839.7	3427.8	6377.3	34237.1	15313.4	10811.0	20120.5	107740.9	48241.3	34143.8	63375.3	29957.7

generalize the current framework to assess the errors of AQP systems for more complex queries, such as join and common aggregation queries, on large datasets.

Acknowledgement

This research is partially supported by University Research Council Grant and Research Professorship Award at Youngstown State University.

References

[1] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. "Join Synopses for Approximate Query Answering", In *Proc. SIGMOD '99*, pp. 275–286, 1999.

[2] S. Agarwal, H. Milner, A. Kleiner, A. Talwalkar, M. Jordan, S. Madden, B. Mozafari, and I. Stoica. "Knowing when You're Wrong: Building Fast and Reliable Approximate Query Processing Systems", In *Proc. SIGMOD 2014*, pp. 481–492, 2014.

[3] K. J. Archer and R. V. Kimes. "Empirical Characterization of Random Forest Variable Importance Measures", *Computational Statistics and Data Analysis*, 52:2249–2260, 2008.

[4] S. Cal, E. Cheng, and F. Yu. "Optimized Bootstrap Sampling for -AQP Error Estimation: A Pilot Study", In *Proc. of ISCA 30th International Conference on Software Engineering and Data Engineering*, 77:144–153, 2021.

[5] S. Chaudhuri, B. Ding, and S. Kandula. "Approximate Query Processing: No Silver Bullet", In *Proc. SIGMOD'17*, pp. 511–519, 2017.

[6] Y. Chen and K. Yi. "Two-Level Sampling for Join Size Estimation", In *Proc. SIGMOD 2017*, ACM, pp. 759–774, 2017.

[7] M. R. Chernick. *Bootstrap Methods: A Guide for Practitioners and Researchers*, John Wiley & Sons, 2008.

[8] T. P. P. Council. "TPC-H Benchmark", `http://www.tpc.org/tpch/`.

[9] C. Doulkeridis and K. Nørvåg. "A Survey of Large-Scale Analytical Query Processing in MapReduce", *The VLDB Journal*, 23:355–380, 6 2014.

[10] B. Efron. "Bootstrap Methods: Another Look at the Jackknife", *The Annals of Statistics*, 7:1–26, 1979.

[11] B. Efron. "Second Thoughts on the Bootstrap", *Statistical Science*, 18:135–140, 2003.

[12] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*, CRC Press, 1994.

[13] A. Kleiner, A. Talwalkar, S. Agarwal, I. Stoica, and M. I. Jordan. "A General Bootstrap Performance Diagnostic", In *Proc. SIGKDD 2013*, pp. 419, 2013.

[14] A. Kleiner, A. Talwalkar, P. Sarkar, and M. Jordan. "The Big Data Bootstrap", *arXiv preprint arXiv:1206.6415*, June 2012.

[15] N. Laptev, K. Zeng, and C. Zaniolo. "Early Accurate Results for Advanced Analytics on MapReduce", *Proc. VLDB Endow.*, 5:1028–1039, June 2012.

[16] V. Leis, B. Radke, A. Gubichev, A. Kemper, and T. Neumann. "Cardinality Estimation Done Right : Index-

based Join Sampling", In *Proc. CIDR'17*, 2017.

[17] F. Li, B. Wu, K. Yi, Z. Zhao, L. Li, S. Miles, Z. Melville, A. Prasad, and L. L. Breeden. "Wander Join: Online Aggregation via Random Walks", In *Proc. SIGMOD'16*, pp. 615–629, 2016.

[18] K. Li and G. Li. "Approximate Query Processing: What is New and Where to Go?", *Data Science and Engineering*, 3:379–397, 2018.

[19] Q. Liu. "Approximate Query Processing", In L. LIU and M. T. ÖZSU, Editors. *Encyclopedia of Database Systems*, Springer US, pp. 113–119, 2009.

[20] A. Pol and C. Jermaine. "Relational Confidence Bounds are Easy with the Bootstrap", In *Proc. SIGMOD'05*, pp. 587–598, 2005.

[21] D. L. Quoc, I. E. Akkus, P. Bhatotia, S. Blanas, R. Chen, C. Fetzer, and T. Strufe. "Approximate Distributed Joins in Apache Spark", *arXiv preprint arXiv:1805.05874* , May 2018.

[22] M. Sch, J. Schildgen, and S. Deßloch. "Sampling with Incremental MapReduce", *Datenbanksysteme für Business, Technologie und Web (BTW 2015)-Workshopband*, 2015.

[23] R. J. Tibshirani and B. Efron. *An Introduction to the Bootstrap*, Monographs on statistics and applied probability, 57:1–436, 1993.

[24] F. Yu, W.-C. Hou, C. Luo, D. Che, and M. Zhu. "CS2: A New Database Synopsis for Query Estimation", ACM, pp. 469–480, 2013.

[25] K. Zeng. "ABS: A System for Scalable Approximate Queries with Accuracy Guarantees", *Sigmod*, pp. 1067–1070, 2014.

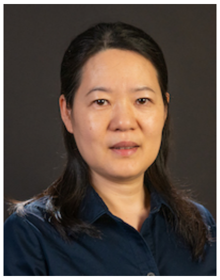
[26] K. Zeng. "Approximation and Search Optimization on Massive Data Bases and Data Streams", PhD Thesis, University of California, Los Angeles, 2014.

[27] Z. Zhou, H. Zhang, S. Li, and X. Du. "Hermes: A Privacy-Preserving Approximate Search Framework for Big Data", *IEEE Access*, 6:20009–20020, 2018.

Feng Yu, Ph.D., is currently an associate professor of Computer Science and Information Systems at Youngstown State University, USA. He is a campus champion for NSF XSEDE. His current research interests include database systems, big data management, and cloud computing. He has served as a reviewer for many international conferences, such as DEXA, SSDBM, and IEEE Big Data, and scholarly journals, such as ACM TODS, Information Sciences, and DKE.

Semih Cal received the B.S. degree from the Department of Computer Science, Sam Houston State University, Texas, USA in 2017 and he received the M.S. degree from the Department of Computer Science, Youngstown State University, Ohio, USA in 2021, He is currently a Ph.D. student in the Department of Computer Science, Texas Tech University. His current research interests include IoT devices, routing protocols in FANET, and mobile data management.

En Cheng, Ph.D., is an associate professor in the Department of Computer Science at The University of Akron. Before joining The University of Akron, Dr. Cheng had the opportunity to experience internships in diverse research centers, including Microsoft Research Asia, IBM T.J. Watson Research Center, and Cleveland Clinic Foundation. Her current research interests include Data Integration, Big Data, Database Systems and Applications, Mobile Applications, Semantic Web, and Business Intelligence.



Lucy Kerns, Ph.D., is currently working as an associate professor in the Department of Mathematics and Statistics at Youngstown State University, where she also serves as the Statistics Coordinator and co-director of the Mathematical and Statistical Consulting Center. Her research interests have been focused mainly on simultaneous inferential techniques, environment risk assessment and environmental toxicology, statistical modeling, data analytics (machine learning and data mining), and data visualization.

Weidong Xiong is a visiting associate lecturer of the Department of Electrical Engineering and Computer Science at Cleveland State University. He received his Ph.D. in Computer Science from Southern Illinois University Carbondale, IL in 2018. His primary research interests include Concurrency Control Protocols in Database, 3D Programming, Volume Rendering, and Deep Learning. Prior to transitioning his career in academia, Xiong was a senior software engineer in IT industry with more than ten years of programming experience.

VR Tracker Location and Rotation Predictions using HTC Vive Tracking System and Gradient Boosting Regressor

Lin Hall, Ping Wang, Grayson Blankenship, Emmanuel Zenil Lopez,
Chris Castro, Zhen Zhu, Rui Wu
East Carolina University, Greenville, NC USA

Abstract

Machine learning and virtual reality technologies have become focal points for research and development in recent years. In this study, we propose a framework to estimate and visualize the position and rotation of human spines, based on predictions made with machine learning. HTC Vive trackers are used to simulate the bone structure. Truth reference data for position and rotation are collected in the HTC system, which are used to evaluate the performance of our solution. Preliminary results show that the pose of the simulated structure can be accurately predicted. The proposed framework can be beneficial to medical training and surgical operations.

Key Words: Tracking, virtual reality, machine learning, gradient boosting regressor

1 Introduction

Rapid advances in technology and medical device development in the 21st century are bringing about a new era of medicine, contributing to healthier and more productive lives. As technology and patient complexity continues to increase, demands for novel approaches to ensure competency have arisen [17].

Machine learning as a means of pattern recognition has been heavily utilized over the last decade. By definition, machine learning is pattern recognition that explains the surrounding environment. Further, this pattern recognition is achieved by modeling human intelligence [7]. In this paper, we utilize machine learning techniques to predict the motion of biomedical systems, such as human spines. Users can visualize the location and orientation of human spines in a virtual reality environment in real time. It provides virtual reality experiences to medical professionals during training, or to serve as an additional visualization and guidance tool during actual surgical operations.

Existing machine learning studies in medical research have been largely focused on clinical datasets and patient diagnoses [3, 15], although some predictive analyses have been attempted in certain areas, including: cancer [4, 14], stroke [11], and dementia [1]. In this work, we explore the real-time prediction capability of machine learning, which is a critical component in image-guided surgery.

During surgical operations, the patient may not be completely motionless. The human body may experience small changes in position and/or orientation. Most existing visualization tools of the human spine rely on mathematical models and imaging equipment [10], which are not capable of handling real-time motion. By contrast, in improved spinal visualization systems [23], machine learning has been successfully used to image the human spine, with quality even rivaling that of manual imaging [9].

On the other hand, virtual reality (VR) is a relatively new concept in medical research. Typically, VR has been considered an educational tool [20]. Virtual reality simulator becomes a powerful tool for surgical trainees to repeatedly practice without potential harm to patients and animals. Traditionally, VR was not widely used in high-fidelity applications. However, recent development in VR technology has enabled higher-accuracy solutions. A 2021 study found that using augmented and virtual reality resulted in 97% accuracy for pedicle screw placement [8]. Unfortunately, this study relied on a static spine, which does not model patient motion during surgery. For example, pedicle screw placement can cause shifts along the spine during a surgical procedure[8]. Therefore, we propose to utilize machine learning techniques to account for such spinal motion.

One of the main applications of machine learning is data visualization [6], particularly so for medical research. The importance of data visualization in terms of knowledge extraction has been documented [21]. Further, when combined with VR, visualization results in a complete immersion into the data [6]. In the proposed system, visualization will be implemented as a three-dimensional immersive VR experience. A section of human spine will be simulated with multiple HTC Vive Pro trackers in this work. Trackers are used to simulate motion of a spine section that is partially rigid. The non-rigid motion of the spine will be predicted and verified with the HTC system. The trackers are visualized using the Unity Virtual Reality Environment in this work.

The main contribution of this paper is to provide a software/hardware framework which integrates VR with machine learning to track, predict and visualize the position and orientation of VR trackers. The framework includes prediction of time series data obtained from the simulated human spine, for which we use a gradient boosting regressor model. Furthermore, we mitigate data outliers by utilizing the extreme event split technique in order to improve the prediction functionality. Finally, the simulated human spine will be visualized in VR. Also, the framework can support other medical visualization applications.

In section 2 of this paper, previous work related to our study is reviewed. In section 3, an introduction to the software and hardware components, architecture of the proposed framework and a process workflow from data collection to virtual reality visualization are presented. The machine learning models and the extreme event split approach are discussed in section 4. Section 5 presents results of spine motion prediction, followed by conclusions.

2 Related Work

The efforts of Bissonnette to distinguish surgical training levels using virtual reality simulator and machine learning methods suggest that virtual reality and machine learning can be powerful tools for surgical training and evaluation. The authors divided spine surgeons, spine fellows, orthopaedic and neurosurgery residents, and medical students from 4 Canadian universities into two groups (senior and junior) according to their training levels. 22 participants were senior and 19 were junior. All the participants were asked to perform a spinal surgery in a virtual reality environment. The virtual hemilaminectomy required participants to remove the L3 lamina with a simulated burr in their dominant hand while controlling bleeding with a simulated suction instrument in their non-dominant hand [2].

Participants were required to remove the L3 lamina in five minutes, without damaging surrounding tissues. Their position, angle, force application of the simulated burr and suction instruments, and removed tissue volumes during the procedure were recorded at 20 ms intervals. The data were collected as metrics for training machine learning algorithms. Five classification algorithms were applied: support vector classifier, K-Nearest Neighbors classifier, Linear Discriminant analysis, Naive Bayes classifier, and decision tree classifier. Regression algorithms can also be combined with virtual reality technique in surgical field. Dubin implemented machine learning algorithms to develop regression models and to predict Global Evaluative Assessment of Robotic Skills(GEARS) score using a VR simulator[5]. GEARS is a validated surgical proficiency testing tool, which has been widely used in training programs. 74 participants were required to perform a basic VR exercise (Ring and Rail1) and a complex VR exercise (Suture Sponge1) on two simulators–dV-Trainer(dVT) and da Vinci Skills Simulator(dVSS). The simulator gave scores of each exercise for each participant. And the recorded video was sent to human subject matter experts for review using the GEARS tool. Linear regression models were generated for each exercise on each simulator to predict GEARS score based on simulator score.

Although both works combined VR with machine learning, they used relatively simple machine learning algorithms, and were focused on the performance from the simulators. Bissonnette et al built classification model using Support Vector Classifier to distinguish senior level and junior level of surgeons. Dubin built simple linear regression to predict the GEAR score of medical students. Besides, the related research built models on relatively small datasets. Bissonnette has forty-one participants for model building and Dubin built linear regression model on 74 participants. The machine learning model required in the spine motion prediction task is a little more complicated. We predict six degrees of freedom in spine motion, three positional variables and three rotational variables. The performance of prediction will be evaluated on all six variables. The input of this model includes thousands of observations. The complexity of the input data requires special handling of extreme values, or data anomalies. An extreme value is an observation at the boundaries of the domain.

Anomaly detection in time series has attracted considerable attention due to its importance in many real-world applications including intrusion detection, energy management The finance [19]. Most anomaly detection methods require manually set thresholds or assumptions on the distribution of data. Isolated forest algorithm is one of the commonly used extreme value detecting algorithms. The term isolation in this case means ”separating an instance from rest of instances”. The two processing stages of isolation forest include training stage and testing stage. The training stage builds isolation trees using subsamples of the training set. The testing stage passes instances in testing set to obtain anomaly score for each instance. In the training stage, isolation trees are constructed by recursively partitioning a subsample X' until all instances are isolated. Each isolation tree is constructed using a subsample X' randomly selected without replacement from X[24].

The normal points tend to be isolated at the deeper end of the tree, whereas anomalies are closer to the tree root, due to their singularity nature. The shorter the average path length, the higher the chances to be anomalies[24]. In teintervsting stage, outliers are identified and labeled based on anomaly score of each instance. The 95% quantile method is used to determine the threshold of extreme value and the instances with anomaly score greater than the threshold are classified as outliers. The extreme value machine (EVM) introduced in 2018, has become an important tool in multivariate statistics and machine learning in the past few years. Generalized Pareto distribution(GPD) classifier is an alternative approach of EVM. It requires the generalized Pareto distribution assumption from extreme value theory. The idea of the EVM is to approximate the distribution of the margin distance of each point in each class using extreme value theory. A new point is then classified as normal if it is inside the margin of some point in the training set with high probability[22]. We applied the DSPOT algorithm of splitting the data into normal events and extreme events. Then we built machine learning models on normal dataset and extreme dataset separately and compared the results with that of model built on dataset without extreme event splitting.

Another challenge in motion prediction lies in the fact that it is essentially a time-series prediction problem. The sliding window technique has been utilized to preserve the temporal relationship of the data [12]. In this technique, the size of the window is of particular importance [12, 13].

3 Methodology

In this work, HTC Vive Pro VR trackers are used to represent parts of the human spine. Two VR tracks simulate rigid-body motion the human spine, whereas a third tracker simulates motion of unknown model. Using a time series regression model, we use location and orientation of the first two trackers to accurately predict the "third" VR tracker. The actual location and orientation of all the trackers can be accurately measured by the VR system, which allows us to assess the accuracy of the machine learning model (R^2, *rmse*, etc.).

The trackers were placed on an HTC Vive Pro racket. To train the machine learning model, motion data including the X, Y, and Z position as well as pitch, roll, and yaw angles were collected using the racket. Next, the motion data used in training were reprocessed, and fed into the time series regression model. After training is completed, prediction accuracy were assessed. Extreme event split and sliding window techniques were implemented to improve the quality of machine learning. Subsequently, this output prediction was fed into a Unity application, such that all three trackers could be displayed in the Unity application scene.

3.1 Architecture

The overall system includes three main components: the VR hardware (HTC Vive Pro trackers and racket), the backend software (machine learning algorithms on a Flask server), and the frontend software (SteamVR and Unity). In this section, we will discuss how the components are connected and interfaced with each other.

Flask is a popular web application framework. In this study, a Flask server is utilized for real-time prediction of the trackers. The processed positional data is input into the server. As a result, when activated, the server will generate predicted positional coordinates of the tracker to be visualized in Unity.

SteamVR is an expansion of the Steam gaming engine that adds a virtual reality component to the gaming experience. In this study, SteamVR is utilized in the data collection and visualization of each tracker. For data collection, SteamVR is utilized to ensure the connection between the HTC Lighthouses (instruments that create the virtual reality boundary) and the HTC VR Headset and Trackers. Further, the data is collected by performing movements in each positional and rotational direction. For visualizing the trackers, SteamVR communicates with Unity to create a scene, or virtual reality environment.

Unity is a real-time development platform, typically used in gaming. However, Unity can be utilized to simulate environments. In this study, Unity is used to visualize the trackers. As mentioned before the visualizations are called scenes.

Figure 1 illustrates the visualization process.

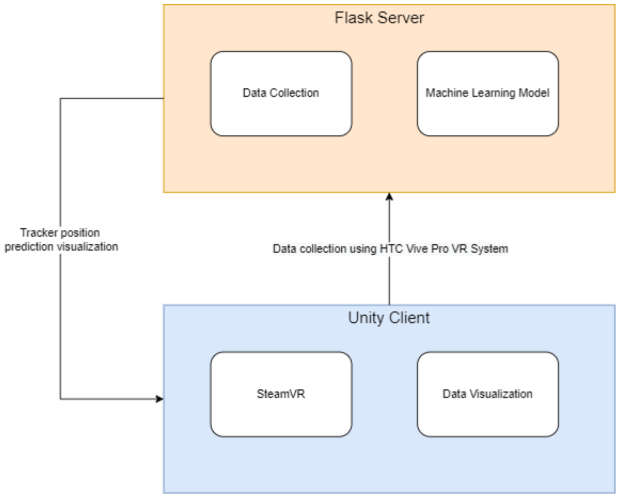


Figure 1: A flowchart illustrating the process in how data collection occurs. Here, the orange section represents components utilizing the Flask server. Likewise, the blue section represents components utilizing the Unity client. First, the data is collected from the HTC Vive Pro Setup. Next, the data is preprocessed and sent to the Flask server for the real-time tracker location prediction. The prediction is generated using a Gradient Boosting Regressor machine learning model. Each HTC Vive Pro Tracker is attached to an HTC Vive Pro Racket, which is considered a rigid body. Finally, using SteamVR and Unity, the tracker locations are visualized

3.2 Data Collection with VR Trackers

Machine learning is heavily dependent on the quality of training data. Therefore, data collection is an integral component to a successful machine learning algorithm, such as gradient boosting regressor shown in Figure 1. The HTC Vive Pro Trackers used in this work have a 270-degree field of view, which allows for data collection in virtually every direction. As afore mentioned, three trackers are manually attached to an HTC Vive Pro Wireless Racket. Each tracker has a unique identifier. The racket allows us to place trackers in a straight line with equidistant positions. On the HTC Vive Pro racket in Figure 1, the predicted tracker is in the middle, and is adjacent on either side by two other trackers. It provides a simplified model of human spine. Although the racket can only simulate linear motion of the third tracker within a rigid body, it is not a requirement for the machine learning algorithm. Non-rigid motion of the tracker can also be predicted.

Sensors attached to the trackers and a HTC Vive Pro Headset must be synced to the Steam virtual reality software, and to the HTC Vive Pro Lighthouses (Base Stations). These wireless lighthouses are responsible for determining the position of the sensors in a VR environment. Typically, these lighthouses are placed approximately six feet apart in a room. The HTC Vive Pro Headset must be active at all times for data collection to occur. Failure to do so will result in the data being inconsistent, or even uncollected.

A Steam virtual reality scene is first initialized, in order to create the required VR environment. In this environment, the raw data are recorded. Unity provides an API to calculate the location of the sensors, which is relative to the lighthouse base stations. The point of origin is selected for a spatial Cartesian XYZ system. When activating the Steam VR environment, it is imperative to have the HTC Vive Pro Racket at the point of origin in order to ensure accurate data collection. Since the sensors are rigidly attached to the trackers and headset, position of multiple sensors on a tracker/headset can then be used to estimate the position and orientation of the whole tracker/headset.

3D position and 3D orientation data (pitch, roll, and yaw angles) are collected. Position and orientation are recorded in the coordinate system defined by the virtual reality environment created using the Unity Game Engine. In the 3D spatial Cartesian coordinate system, X and Y are horizontal axes and Z is the vertical axes. However, X and Z in the Unity environment are horizontal, corresponding to the Cartesian X and Y directions respectively. The Unity Y axis is vertical, equivalent to the Cartesian Z direction.

While the Steam VR environment scene is active, the output of data occurs continuously until the scene is stopped. During each step of the data collection, we focus on only one of the dimensions. The HTC Vive Pro Racket (with each tracker attached) is shifted in the desired dimension at different positions. For example, if data collection is focused on the Unity X direction, the racket motion will be primarily on the X direction. The motion on X direction will be random, and will be repeated at various positions. At each position, data collection lasts approximately 15-20 seconds. Subsequently, a dataset is created for each of the six dimensions. Therefore, there will be six individual data files.

The recorded data are preprocessed to remove incomplete samples and null values, and subsequently recorded in comma-separated value (csv) format. The csv files are cleaned to remove redundant information. For instance, if data is collected for the X position, the Y and Z positional information is removed from this file. As a result, the data are less noisy, purely focused on one direction at a time. It allows for direct observability in each of the dimensions.

Preprocessed data files are tested for quality. Here, simple linear regression is used from the scikit-learn machine learning library [16]. To ensure the accuracy of the data collection for each directional file, the R^2 value is calculated. If the dataset was incomplete, or held any null or unaccepted data types, a modeling error is returned. Further, if the R^2 value was low, the data file will be recollected.

3.3 Extreme Event Split

We applied Drift Streaming Peaks-Over-Threshold(DSPOT) to detect extreme events of the time series data and split the dataset into normal dataset and extreme dataset. As stated earlier, isolation forest extreme value detector and GPD classifier rely on either manually set thresholds or assumptions on the distribution of data. By using the DSPOT approach, we do not assume the distribution of the value but rely on extreme value theory to estimate accurately low probability areas and then discriminate outliers [19].

3.4 Real-time Prediction

The preprocessed data is sent to a Flask Server. On this server, a gradient boosting regressor machine learning model is implemented. The predicted location and orientation of the tracker are updated and distributed continuously as long as the server is active. The prediction is then sent to the Unity application and are visualized. Figure 2 shows the visualization of the trackers, including the predicted location.

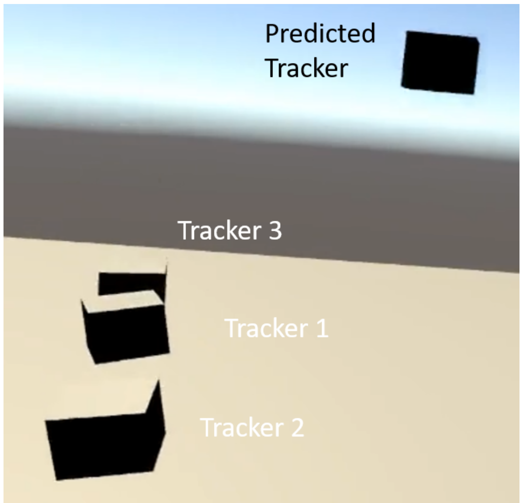


Figure 2: An illustration of the virtual reality visualization of the HTC Vive Pro Wireless Racket setup with the attached trackers. During surgery, sensors are placed on the spine, trackers 2 and 3 are external sensors, and tracker 1 is for bone location. When predicted correctly, the visualized tracker should overlay tracker 1

3.5 Window Slider Technique

Instead of predicting the target variable using the whole training dataset, the window slider technique helps improve the accuracy of predictive models by capturing the most complete information possible from the dataset.

Figure 3 [18]shows how the sliding window technique reshapes the information by windows with fixed size. The X axes shows the time, and the y axes shows the response variable. Predictors are not shown in this figure. The window size of this example is 4, which means that the model is going to map the 4 observations in this window and predict the value at time

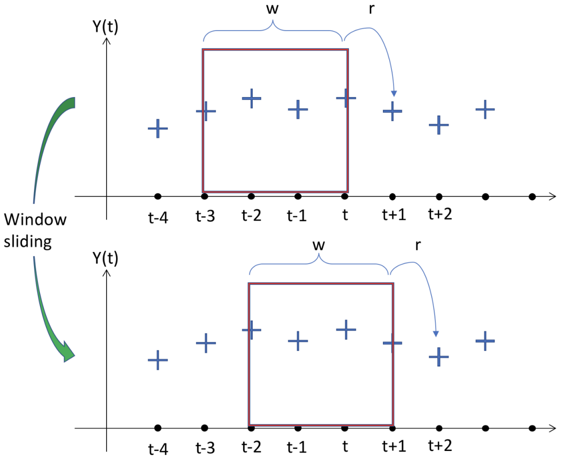
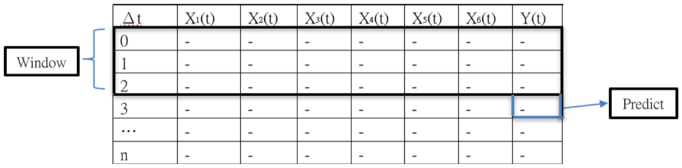


Figure 3: Window slider

t+1. Then the sliding window moves forward one time step and a response variable at t+2 is predicted. The sliding window continues moving forward and proceeding the same prediction step until the end of the time series dataset.

Instead of predicting one variable in a time series dataset as shown above, the tracker dataset we applied sliding window technique to has six predictor variables–rotational X,Y,Z variables of tracker one and two, and one response variable–rotational X variable. We have n-w windows in total, where n representing the sample size of training set, and w representing the window size. Sliding window technique was applied to machine learning models and RMSE values will be calculated to evaluate the accuracy of models.

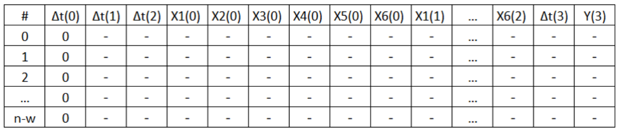
Figure 4 shows an example of defining windows in the tracker dataset with window size set to 3. The data records in the black frame are a training set used to train the models. There are seven predictor variables in this training set–Δt and X1-X6, and one response variable–Y. The response variable next to the window is predicted with the trained model. This procedure is repeated n-w times as the window slides down one row each time.



Δt	$X_1(t)$	$X_2(t)$	$X_3(t)$	$X_4(t)$	$X_5(t)$	$X_6(t)$	Y(t)
0	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-
···	-	-	-	-	-	-	-
n	-	-	-	-	-	-	-

Figure 4: Window with size=3

Figure 5 shows the new dataset generated using the sliding window technique with window size set to 3. Models are trained using this new dataset and RMSE values are calculated to evaluate the model. This procedure is repeated with different sample sizes.



#	Δt(0)	Δt(1)	Δt(2)	X1(0)	X2(0)	X3(0)	X4(0)	X5(0)	X6(0)	X1(1)	...	X6(2)	Δt(3)	Y(3)
0	0	-	-	-	-	-	-	-	-	-	...	-	-	-
1	0	-	-	-	-	-	-	-	-	-	...	-	-	-
2	0	-	-	-	-	-	-	-	-	-	...	-	-	-
...	0	-	-	-	-	-	-	-	-	-	...	-	-	-
n-w	0	-	-	-	-	-	-	-	-	-	...	-	-	-

Figure 5: New dataset

To determine the window size, autocorrelation is utilized. Autocorrelation represents the degree of similarity between a time series by measuring the relationship between a variable's current values and its historical values over successive time intervals[25]. Eq. 1 is used to calculate the autocorrelation.

$$\hat{\rho_k} = \frac{\Sigma_{t=k+1}^{T}(r_t - \bar{r})(r_{t-k} - \bar{r})}{\Sigma_{t=1}^{T}(r_t - \bar{r})^2} \tag{1}$$

4 Results

As stated earlier, the gradient boosting regressor model, along with the extreme event split and the sliding window technique, were utilized in this study. Table 1 shows the results of each model run for each position and rotation.

4.1 Gradient Boosting Regressor

Initial results of the gradient boosting regressor model show a high accuracy with low root mean square error for each position (X, Y, Z) and pitch rotation. However, roll and yaw did not produce low-error results. This high accuracy is high due to the linearity, as well as continuity of the data. The lower accuracies in roll and yaw can be attributed to the non-linearity of the data, where a pitch movement is more linear than roll or yaw movements. Table 1 shows the accuracy of the predicted tracker locations for each position and rotation in relation to the actual tracker coordinates. As a whole, the GBR performed well.

The extreme event split and sliding window technique performed better than the base gradient boosting regressor model. This indicates the data contained several outliers, and that the base model does not perform as well when the data is inspected all at once. Also, this is important since the data collection is performed by manual simultaneous movements of the trackers. Therefore, it is possible not all of the movements are consistent.

Consequently, the better comparison is between the extreme event split and the sliding window technique. The extreme event split performed better overall for positional movements, while the sliding window technique performed better for rotational movements. This is a result of the extreme event split accounting for the outliers in the mostly linear positional dataset. However, the sliding window technique performs better with the rotational data due to its multiple examinations of the dataset, which are correctly considered windows. Further, the rotational data is the least linear of the collected data.

Table 1: This table displays the model results of the base gradient boosting regressor model prediction, the extreme-event split, and the sliding window technique on the input data. Root Mean Square Error was calculated for each type of model run. The GBR model performed well for each position and rotation, with the exception of the pitch rotation. The extreme event split was better for the X, Y, and Z positions, while the sliding window technique performed best for the pitch, roll, and yaw rotations

Method	Metric	Position					
		X	Y	Z	Pitch	Roll	Yaw
GBR (Base Data)	RMSE	0.100	0.117	0.204	0.444	848.046	515.981
GBR (Extreme Event Split)	RMSE	0.036	0.036	0.036	0.353	18.266	11.673
GBR (Sliding Window)	RMSE	0.313	0.260	0.313	0.250	14.312	9.366

5 Conclusion

In this paper, we proposed a framework to visualize and predict HTC Vive trackers. The experimental results show that our proposed method is promising and can be possibly applied to medical usage. Here, the extreme event split proved best for improving the model results for the X, Y, and Z positions. Conversely, the sliding window technique performed best for the pitch, roll, and yaw rotations.

In Table 1, we have shown the results of the predicted tracker location in relation to tracker 1 for this research. As a result, the gradient boosting regressor model has proven useful for the machine learning application of this research, as well as the utilization of the extreme-event split and sliding window techniques.

With that said, some improvements can be made to ensure further decreases in error within the models. Parameter tuning of the extreme-event split and sliding window technique would further improve accuracy within the dataset. Also, the overarching goal of this research is to visualize tracker locations. By utilizing a mixture of the techniques, a better visualization would be possible. More specifically, utilizing the extreme event split for the positional data, and sliding window technique for the rotational data should produce a more accurate visualization.

Future research goals include utilizing multiple techniques, as mentioned above. Also, prediction for a non-rigid body is significant for an actual human body.

6 Acknowledgements

This work is supported in part by the National Science Foundation IUSE/PFE:RED award #1730568.

References

[1] Gopi Battineni, Nalini Chintalapudi, and Francesco Amenta. Machine Learning In Medicine: Performance Calculation of Dementia Prediction by Support Vector Machines (SVM). *Informatics in Medicine Unlocked*, 16:100200, 2019.

[2] Vincent Bissonnette, Nykan Mirchi, Nicole Ledwos, Ghusn Alsidieri, Alexander Winkler-Schwartz, and Rolando F Del Maestro. Artificial Intelligence Distinguishes Surgical Training Levels in a Virtual Reality Spinal Task. *The Journal of Bone and Joint Surgery: American volume*, 101(23):e127, 2019.

[3] Deo Rahul C. Machine Learning in Medicine. *Circulation*, 132(20):1920–1930, Nov 2015.

[4] Joseph A. Cruz and David S. Wishart. Applications of Machine Learning in Cancer Prediction and Prognosis. *Cancer Informatics*, 2:117693510600200030, Jan 2006.

[5] Ariel Kate Dubin, Danielle Julian, Alyssa Tanaka, Patricia Mattingly, and Roger Smith. A Model for Predicting the GEARS Score from Virtual Reality Surgical Simulator Metrics. *Surgical Endoscopy*, 32(8):3576–3581, 2018.

[6] Mohamed El Beheiry, Sébastien Doutreligne, Clément Caporal, Cécilia Ostertag, Maxime Dahan, and Jean-Baptiste Masson. Virtual Reality: Beyond Visualization. *Journal of Molecular Biology*, 431(7):1315–1321, 2019.

[7] Issam El Naqa and Martin J. Murphy. *What Is Machine Learning?*, page 3–11. Springer International Publishing, 2015.

[8] Mitchell S. Fourman, Hamid Ghaednia, Amanda Lans, Sophie Lloyd, Allison Sweeney, Kelsey Detels, Hidde Dijkstra, Jacobien H.F. Oosterhoff, Duncan C. Ramsey, Synho Do, and Joseph H. Schwab. Applications of Augmented and Virtual Reality in Spine Surgery and Education: A Review. *Seminars in Spine Surgery*, page 100875, 2021.

[9] Fabio Galbusera, Gloria Casaroli, and Tito Bassani. Artificial Intelligence and Machine Learning in Spine Research. *JOR SPINE*, 2(1):e1044, 2019.

[10] L. Humbert, J.A. De Guise, B. Aubert, B. Godbout, S. Parent, D. Mitton, and W. Skalli. 3D Reconstruction of the Spine From Biplanar X-Rays Using Longitudinal and Transversal Inferences. *Journal of Biomechanics*, 40:S160, 2007. Program and Abstracts of the XXI Congress, International Society of Biomechanics.

[11] C. Hung, W. Chen, P. Lai, C. Lin, and C. Lee. Comparing Deep Neural Network and Other Machine Learning Algorithms For Stroke Prediction in a Large-Scale Population-Based Electronic Medical Claims Database. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3110-3113, Jul 2017.

[12] WH Wan Ishak, Ku-Ruhana Ku-Mahamud, and Norita Md Norwawi. Mining Temporal Reservoir Data Using Sliding Window Technique. *CiiT International Journal of Data Mining Knowledge Engineering*, 3(8):473–478, 2011.

[13] Hua-Fu Li and Suh-Yin Lee. Mining Frequent Itemsets Over Data Streams Using Efficient Window Sliding Techniques. *Expert Systems With Applications*, 36(2):1466–1477, 2009.

[14] Yixuan Li and Zixuan Chen. Performance Evaluation of Machine Learning Methods for Breast Cancer Prediction. *Applied and Computational Mathematics*, 7(4):212, 2018.

[15] Ziad Obermeyer and Ezekiel J. Emanuel. Predicting the Future — Big Data, Machine Learning, and Clinical Medicine. *The New England Journal of Medicine*, 375(13):1216–1219, Sep 2016.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] Michael P Rogers, Anthony J DeSantis, Haroon Janjua, Tara M Barry, and Paul C Kuo. The Future Surgical Training Paradigm: Virtual Reality and Machine Learning in Surgical Education. *Surgery*, 2020.

[18] Pablo Ruiz. ML Approaches for Time Series. `https://towardsdatascience.com/ml-approaches-for\\-time-series-4d44722e48fe`. 2019.

[19] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly Detection in Streams With Extreme Value Theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1067–1075, 2017.

[20] Samuel B. Tomlinson, Benjamin K. Hendricks, and Aaron Cohen-Gadol. Immersive Three-Dimensional Modeling and Virtual Reality for Enhanced Visualization of Operative Neurosurgical Anatomy. *World Neurosurgery*, 131:313–320, 2019.

[21] Alfredo Vellido. The Importance of Interpretability and Visualization in Machine Learning for Applications in Medicine and Health Care. *Neural Computing and Applications*, pages 1–15, 2019.

[22] Edoardo Vignotto and Sebastian Engelke. Extreme Value Theory for Anomaly Detection–The GPD Classifier. *Extremes*, 23(4):501–520, 2020.

[23] Voinea, Gheorghe-Daniel and Butnariu, Silviu and Mogan, Gheorghe. Measurement and Geometric Modelling of Human Spine Posture for Medical Rehabilitation Purposes Using a Wearable Monitoring System Based on Inertial Sensors. *Sensors*, 17(1):1–19, 2017.

[24] Luqing Wang, Qinglin Zhao, Si Gao, Wei Zhang, and Li Feng. A New Extreme Detection Method for Remote Compound Extremes in Southeast China. *Frontiers in Earth Science*, 9:243, 2021.

[25] Zach. How to Calculate Autocorrelation in Python. `https://www.statology.org/autocorrelation-python`. 2020.

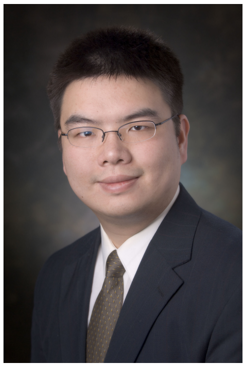
Lin Hall is a candidate for the Master of Science in Data Science at East Carolina University, which he is on track to complete in May 2022. His main research interests involve machine learning and artificial intelligence and their utilization in AR/VR visualization. He also holds interests in automated systems and robotics, as well as atmospheric science. His research on this has resulted in one journal publication, but has also published two journals in the field of atmospheric science. He received finalist for Best Paper for the 2021 SEDE Conference, as well winning Best Master's Paper from the Southeastern Division of the Association of American Geographers in 2012. His research in atmospheric science also earned him the NASA Space Grant.

Ping Wang graduated in December 2021 with a Master of Science in Computer Science from ECU. Her research interest is machine learning. She currently works as a Data Analyst in Pennsylvania.

Grayson Blankenship graduated in May 2021 from East Carolina University with a Bachelor's in Computer Science. His research interests were with augmented and virtual reality, machine learning, and video games. He was supervised by Dr. Rui Wu. Grayson currently works as a QA Game Tester for Epic Games, and has held this position since Summer 2021.

Emmanuel Zenil Lopez graduated in May 2020 from East Carolina University with a Bachelor's in Computer Science. His research interests were with augmented and virtual reality, machine learning, and video games. He was supervised by Dr. Rui Wu.

Chris Castro graduated in May 2020 from East Carolina University with a Bachelor's in Computer Science. His research interests were with augmented and virtual reality, machine learning, and video games. He was supervised by Dr. Rui Wu.



Zhen Zhu is an associate professor at the Department of Engineering. Before joining ECU, he was a senior research engineer and a principal investigator with the Navigation Systems Division and the Advanced Concepts and Technologies Division in Northrop Grumman Electronic Systems from 2010 to 2013. From 2006 to 2010 he worked for the Ohio University Avionics Engineering Center as a senior research engineer, and he was also an adjunct assistant professor with the School of Electrical Engineering and Computer Science in Ohio University. He has developed various types of systems and algorithms for autonomous navigation and guidance of manned and unmanned aircraft; artificial intelligence and software radio.

Rui Wu received a Bachelor's degree in Computer Science and Technology from Jilin University, China in 2013. He then went on and received his Master and Ph.D. degrees in Computer Science and Engineering from the University of Nevada, Reno in 2015 and 2018, respectively. Rui is now working as an assistant professor in the Department of Computer Science at East Carolina University and collaborates with geological and hydrological scientists to protect the ecological system. His main research interests are machine learning and data visualization using AR/VR devices.

Journal Submission

The International Journal of Computers and Their Applications is published four times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

A. Procedure for Submission of a Technical Paper for Consideration

1. Email your manuscript to the Editor-in-Chief, Dr. Ajay Bandi. Email: ajay@nwmissouri.edu.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.
4. **Note**: Papers shorter than 10 pages long will be returned.

B. Manuscript Style:

1. **WORD DOCUMENT**: The text should be **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages. Or it can be single spaced double column.

   LaTex DOCUMENT: The text is to be a double column (10 point font) in pdf format.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month and year.
5. The figures are to be integrated in the text after referenced in the text.

C. Submission of Accepted Manuscripts

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-Chief. If one wished to use LaTex, please see the corresponding LaTex template.
2. The submission may be on a CD/DVD or as an email attachment(s). **The following electronic files should be included:**
   - Paper text (required).
   - Bios (required for each author).
   - Author Photos are to be integrated into the text.
   - Figures, Tables, and Illustrations. These should be integrated into the paper text file.
3. Reminder: The authors photos and short bios should be integrated into the text at the end of the paper. All figures, tables, and illustrations should be integrated into the text after being mentioned in the text.
4. The final paper should be submitted in (a) pdf AND (b) either Word or LaTex. For those authors using LaTex, please follow the guidelines and template.
5. Authors are asked to sign an ISCA copyright form (http://www.isca-hq.org/j-copyright.htm), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

Publication Charges

After a manuscript has been accepted for publication, the contact author will be invoiced a publication charge of **$500.00 USD** to cover part of the cost of publication. For ISCA members, publication charges are **$400.00 USD** publication charges are required.

Revised 2020

ISCA INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS Vol. 29, No. 1, Mar 2022