BIGtensor: Mining Billion-Scale Tensor Made Easy

Namyong Park*
Seoul National University
namyong.park@snu.ac.kr

Byungsoo Jeon*
Seoul National University
jbsimdicd@gmail.com

Jungwoo Lee
Seoul National University
ljw9111@snu.ac.kr

U Kang
Seoul National University
ukang@snu.ac.kr

ABSTRACT

Many real-world data are naturally represented as tensors, or multi-dimensional arrays. Tensor decomposition is an important tool to analyze tensors for various applications such as latent concept discovery, trend analysis, clustering, and anomaly detection. However, existing tools for tensor analysis do not scale well for billion-scale tensors or offer limited functionalities.

In this paper, we propose BIGtensor, a large-scale tensor mining library that tackles both of the above problems. Carefully designed for scalability, BIGtensor decomposes at least 100× larger tensors than the current state of the art. Furthermore, BIGtensor provides a variety of distributed tensor operations and tensor generation methods. We demonstrate how BIGtensor can help users discover hidden concepts and analyze trends from large-scale tensors that are hard to be processed by existing tensor tools.

Keywords

Tensor, Tensor Decompositions, Distributed Computing

1. INTRODUCTION

Many real-world data with multiple attributes are naturally represented as tensors, or multi-dimensional arrays. Examples include movie rating data (movie, user, time, rating) and network intrusion logs (source IP, destination IP, port number, time), to name a few.

Tensor decomposition is a crucial tool that allows us to analyze such data for various applications including latent concept discovery, trend analysis, clustering, and anomaly detection [11]. PARAFAC and Tucker are two widely-used tensor decomposition methods [9]. Each method comes with two different versions: the unconstrained and the nonnegativity-constrained. The nonnegative tensor decomposition enforces all elements in the factors to be nonnegative. Sometimes we have supplementary information coupled with certain modes of the tensor data. For example, for the (movie, user, time, rating) tensor, we may have a (movie, genre) matrix. Coupled Matrix-Tensor Factorization (CMTF) [1] is the standard tool to jointly analyze coupled matrix-tensor data.

*These authors contributed equally to this work.

CIKM '16, October 24–28, 2016, Indianapolis, IN, USA.
© 2016 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4073-1/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2983323.2983332

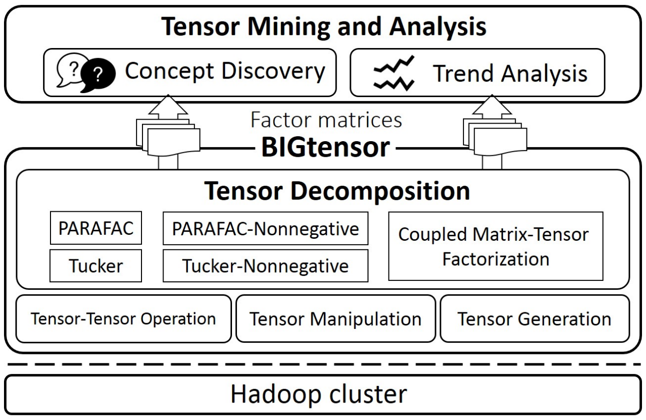


Figure 1: Overview of the system which consists of two parts: BIGtensor is a suite of large-scale algorithms running on the Hadoop platform to handle tensor and tensor decomposition; the tensor mining and analysis module allows the user to discover concepts and analyze trends from the factor matrices, which are the results of tensor decomposition.

Although there exist several tools for analyzing tensors [3, 2, 4, 13, 1], they suffer from either or both of the following limitations:

1. Limited scalability. They run on a single machine, and thus cannot handle large tensors that exceed the capacity of a single machine. While distributed tools also exist (e.g. [4]), their scalability is still limited for truly large tensors [5].
2. Limited functionality. They do not provide a unified framework to handle tensors and tensor decompositions. They support only a couple of tensor algorithms, requiring the user to assemble several different tools to process tensors. Table 1 shows a comparison between BIGtensor and two other state-of-the-art tensor tools.

In this paper, we present BIGtensor, a large-scale tensor mining library running on the Hadoop platform. As a unified framework to handle tensor and tensor decompositions, BIGtensor provides four different sets of functionalities:

- **Tensor Decomposition.** PARAFAC, Nonnegative PARAFAC, Tucker, Nonnegative Tucker, CMTF.
- **Tensor Generation.** FromFactors (from factor matrices and core tensor), Random (tensor values are set randomly), etc.
- **Tensor-Tensor Operation.** BinaryOps (and, or, xor),

ArithmeticOps (+, −, ×, ÷), NModeProduct (n-mode product of two tensors), etc.

- **Tensor Manipulation.** Matricization (unfolding a tensor into a certain mode), Convert2Binary (converting a tensor into a binary tensor), etc.

Carefully designed for scalability, BIGtensor decomposes at least 100× larger tensors than the current state of the art.

For our demonstration, we apply BIGtensor to large real-world tensors that lie beyond the ability of existing tools. We will show how we can employ BIGtensor for tensor decomposition and analysis, and invite the audience to uncover latent concepts and trends from real-world tensor data on their own by interactively exploring the factor matrices decomposed from the Microsoft academic graph, NELL, DARPA, MovieLens, YELP, and PhoneCall tensor data (see Table 3 for details on these tensors). The web site for this paper is located at http://datalab.snu.ac.kr/bigtensordemo.

Table 1: Comparison of functionalities provided by BIGtensor and other state-of-the-art tensor tools. P: PARAFAC, T: Tucker, PN: PARAFAC-Nonnegative, TN: Tucker-Nonnegative, C: CMTF.

Functionality	BIGtensor	Tensor Toolbox	Flexi-FaCT
Tensor Decomposition	**P, PN, T, TN, C**	P, PN, T	P, PN, C
Tensor Generation	**Yes**	Yes	No
Tensor-Tensor Operation	**Yes**	Yes	No
Tensor Manipulation	**Yes**	Yes	No
Distributed	**Yes**	No	Yes

2. SYSTEM OVERVIEW

Our system consists of two parts: 1) BIGtensor and 2) the tensor mining and analysis module. As illustrated in Figure 1, our system relies on BIGtensor for the decomposition of large-scale tensors, and uses the tensor mining and analysis module to discover concepts and analyze trends from the resulting factor matrices. In the following subsections, we give an overview of our system.

2.1 Distributed Tensor Decomposition

BIGtensor supports both tensor decomposition and coupled matrix-tensor decomposition (CMTF) methods. A common challenge shared by both types of decompositions is to scale up to large tensors with millions or billions of elements in each dimension, that are becoming more common in real-world settings. The most important issue is the 'intermediate data explosion' problem [8] where the amount of space required for the materialization of intermediate matrices is too large for a single machine, or even for a cluster.

BIGtensor is carefully designed for massive scalability. The main ideas [8, 7, 6, 5] employed by BIGtensor are as follows: 1) BIGtensor solves the intermediate data explosion problem by (a) exploiting the sparsity of real-world tensors, (b) decoupling the steps involved in n-mode vector product, (c) reusing intermediate data, and (d) using optimizations for MapReduce such as parallel outer products and distributed cache multiplication. 2) BIGtensor minimizes floating point operations through careful ordering of computations to update tensor factors. 3) BIGtensor significantly reduces the number of jobs and disk accesses by integrating redundant MapReduce jobs. Due to the lack of space, we refer the reader to [8, 7, 6, 5] for more details on these ideas.

Scalability. We demonstrate the scalability of tensor decomposition using BIGtensor by comparing BIGtensor against the two state-of-the-art tools, the Tensor Toolbox [3] and FlexiFaCT [4], in terms of tensor size. We run BIGtensor and FlexiFaCT on a Hadoop cluster with 40 machines, each of which has a quad-core Intel Xeon E3-1230v3 CPU and 32GB RAM. The Tensor Toolbox runs on a single machine in the cluster. In all experiments, the rank size for PARAFAC is set to 10, and the core tensor size for Tucker is set to 10×10×10.

Table 2: Scalability of BIGtensor and other tools. We report the mode length and the density of the largest data each tool processes using two representative tensor decomposition algorithms, PARAFAC and Tucker. BIGtensor decomposes 100× larger data in terms of mode length than both of the tools, and also decomposes 100× denser data than the Tensor Toolbox.

Scalability	Method	BIG tensor	Tensor Toolbox	Flexi-FaCT
Mode Length & Nonzeros	PARAFAC	$\geq 10^9$	$\leq 10^7$	$\leq 10^7$
	Tucker	$\geq 10^9$	$\leq 10^7$	–
Density	PARAFAC	$\geq 10^{-5}$	$\leq 10^{-7}$	$\geq 10^{-5}$
	Tucker	$\geq 10^{-5}$	$\leq 10^{-7}$	–

We measure the scalability of each tool on synthetic random tensors in the following two aspects. (a) *Mode Length and Number of Nonzeros.* We increase the length $I = J = K$ of each mode from 10^3 to 10^9, and set the number of nonzeros to $I \times 10$. In Table 2, we report the size of the largest data each tool processes using two representative decomposition algorithms, PARAFAC and Tucker. While BIGtensor processes 10^9 scale tensor, both the Tensor Toolbox and FlexiFaCT run out of memory when the mode length goes beyond 10^7. (b) *Density.* We increase the tensor density from 10^{-9} to 10^{-5}, while fixing the mode length $I = J = K$ to 10^5. Table 2 shows that BIGtensor processes 100× denser data than the Tensor Toolbox. For both of the above aspects, nonnegative PARAFAC and Tucker, and CMTF also show a similar performance.

2.2 Tensor Mining and Analysis

Tensor decomposition enables us to analyze the original tensor data from multiple perspectives. By analyzing factor matrices, which are the output of tensor decomposition, we discover latent concepts, anomalies, and trends.

However, for large-scale tensors, two challenges exist with this process. First, it is infeasible to investigate all elements of factor matrices since the number of elements in each factor matrix is the same as that of the huge input tensor. For example, for NELL data, the minimum number of elements to explore in each factor matrix is 26 million. To handle this issue, we consider only the top-k highest valued elements in each factor matrix. Second, even after this top-k filtering, there are often elements that appear in most columns of the factor matrix. Although their values are high, they do not add much information. Therefore, we calculate new scores for the top-k elements to reflect the specificity of elements that appear in only a few columns. New scoring function based on IDF score is $(1 + \alpha \log(R/col)) \times val$, where α is a predefined constant, R is the number of columns in the factor matrix, col is the number of columns the element appears in, and val is the value of the element. We set α to 2.0. Then, we sort the elements in the descending order of these new scores.

Concept Discovery. Columns of the factor matrices can be roughly regarded as the soft-clustering of the tensor data. Examining each preprocessed factor matrix and seeing what kind of elements form each column reveals the nature of concepts hidden in the data. For instance, we discover 'Comedy' and 'Horror' concepts from MovieLens data as in Figure 4. Furthermore, elements whose values are distinct from those of other elements in the same concept represent anomalies. Figure 5 shows that we find an attacker whose value is the highest in the first concept. To facilitate the finding of concepts, we can also use additional information for each attribute, such as the user demographics. Visualizing additional information for the selected concept shows its various features at a single glance.

Trend Analysis. For tensors with a timestamp attribute, we can discover temporal trends from the time factor matrix. Since vectors for the n-th concept from different factor matrices are related, a temporal pattern indicates a temporal characteristic of the associated factors. For example, we find from MovieLens data that the 'Horror' concept experiences seasonal spikes as shown in Figure 4. For trend analysis, we visualize the temporal values of each concept over time, and look for interesting patterns and temporal correlations between concepts.

3. DEMONSTRATION PLAN

Demonstration Detail. In our demonstration, the audience will be invited to apply PARAFAC and Tucker to several real-world datasets, and analyze them using our system. We will prepare three ways for the users to learn about our system. First, we will prepare a poster to give an overview of the system. Second, we will give a guided tour in which we show the process of how we arrived at our discoveries from the data. Third, we will let our audience to mine the data and make their own discoveries with our system.

Users will be given the options to select (1) the tensor data, (2) the tensor decomposition method among PARAFAC and Tucker, (3) the rank size for PARAFAC or the core tensor size for Tucker, and (4) the analysis mode between concept discovery and trend analysis.

For concept discovery, we provide an overview mode and a detail mode. In the overview mode, users will be able to see all vectors in a specific factor they choose, such as the movie factor, and their top-k elements in a tabular form. More details on each element will appear when the user moves the mouse over it. In the detail mode (Figure 2), they will have a more comprehensive look at each vector of a particular factor matrix. Users will be presented highly ranked elements of the chosen vector. Additional information for each element, such as the movie genres and its statistics, will also be displayed when available to help with the discovery process of the audience. Since vectors for the n-th concept from different factor matrices are closely related with each other, we also show the other vectors of the n-th concept from other factor matrices on the side for a better understanding of the characteristics of the n-th concept in general.

In the trend analysis mode (Figure 3), we visualize the values of temporal elements (e.g. 'year-month' elements) over time so that the user can find meaningful patterns. The tool supports overlapping multiple graphs; users can choose to visualize several concepts at the same time, and look for temporal relationships among them. The graph is zoomable; users can zoom into a shorter time frame, for example, to observe the pattern of a concept over the course of a day, or they can zoom out to month or year level to have a broader view. Moreover, our tool displays those vectors from other factor matrices, which correspond to the concepts selected by the user. This makes it convenient to associate the temporal characteristic with the linked dimension at a glance.

Datasets. For the demonstration, we will use the following real-world tensors: Microsoft academic graph, NELL, DARPA, MovieLens, YELP, and PhoneCall (see Table 3).

4. RELATED WORK

Several scalable algorithms for tensor analysis have been developed. GigaTensor [8] is the first work on large-scale PARAFAC decomposition running on MapReduce. HaTen2 [7, 6] improves upon GigaTensor, and combines Tucker and PARAFAC decompositions into a general framework. Jeon et al. [5] propose SCouT for scalable coupled matrix-tensor factorization. BIGtensor unifies tensor decomposition methods proposed by [8, 7, 5] in a single library, and further improves PARAFAC and Tucker decomposition algorithms from [8, 7] by adopting the ideas from [5] with code optimization. As a tensor mining library, BIGtensor also provides various tensor operations and tensor generation methods. Shin et al. [12] propose PARAFAC-based distributed tensor decomposition methods for partially observable tensors. In [4], Beutel et al. propose FlexiFaCT, a scalable MapReduce algorithm for decomposing matrix, tensor, and coupled matrix-tensor through distributed stochastic gradient descent. ParCube [10] is a parallelizable PARAFAC decomposition method that leverages random sampling techniques.

5. ACKNOWLEDGMENTS

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning (grant No. 2013R1A1A1064409). This work was also supported by ICT R&D program of MSIP/IITP. [R0101-15-0176, Development of Core Technology for Human-like Self-taught Learning based on Symbolic Approach]

6. REFERENCES

[1] E. Acar, T. G. Kolda, and D. M. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. *CoRR*, abs/1105.3422, 2011.

[2] C. A. Andersson and R. Bro. The n-way toolbox for matlab. *Chemometrics and Intelligent Laboratory Systems*, 52(1):1–4.

[3] B. W. Bader, T. G. Kolda, et al. Matlab tensor toolbox version 2.6. Available online, February 2015.

[4] A. Beutel, P. P. Talukdar, A. Kumar, C. Faloutsos, E. E. Papalexakis, and E. P. Xing. Flexifact: Scalable flexible factorization of coupled tensors on hadoop. In *SDM*, 2014.

[5] B. Jeon, I. Jeon, S. Lee, and U. Kang. Scout: Scalable coupled matrix-tensor factorization-algorithms and discoveries. In *ICDE*, 2016.

[6] I. Jeon, E. E. Papalexakis, C. Faloutsos, L. Sael, and U. Kang. Mining billion-scale tensors: algorithms and discoveries. *The VLDB Journal*, 25(4):519–544, 2016.

[7] I. Jeon, E. E. Papalexakis, U. Kang, and C. Faloutsos. Haten2: Billion-scale tensor decompositions. In *ICDE*, 2015.

[8] U. Kang, E. E. Papalexakis, A. Harpale, and C. Faloutsos. Gigatensor: scaling tensor analysis up by 100 times - algorithms and discoveries. In *KDD*, pages 316–324, 2012.

[9] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[10] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos. Parcube: Sparse parallelizable tensor decompositions. In *ECML-PKDD*, pages 521–536, 2012.

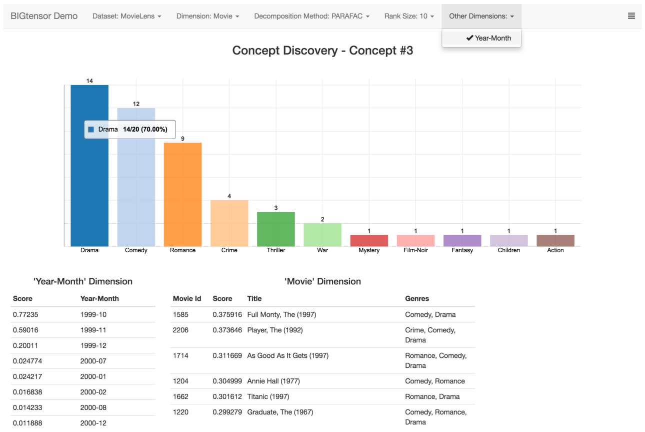
[11] L. Sael, I. Jeon, and U. Kang. Scalable tensor mining. *Big Data Research*, 2(2):82 – 86, 2015. Visions on Big Data.

[12] K. Shin and U. Kang. Distributed methods for high-dimensional and large-scale tensor factorization. In *ICDM*, 2014.

[13] J. O. Yoo, A. Ramanathan, and C. J. Langmead. Pytensor: A python based tensor library. Technical Report CMU-CS-10-102, Carnegie Mellon University, 2010.

APPENDIX

A. SCREENSHOTS



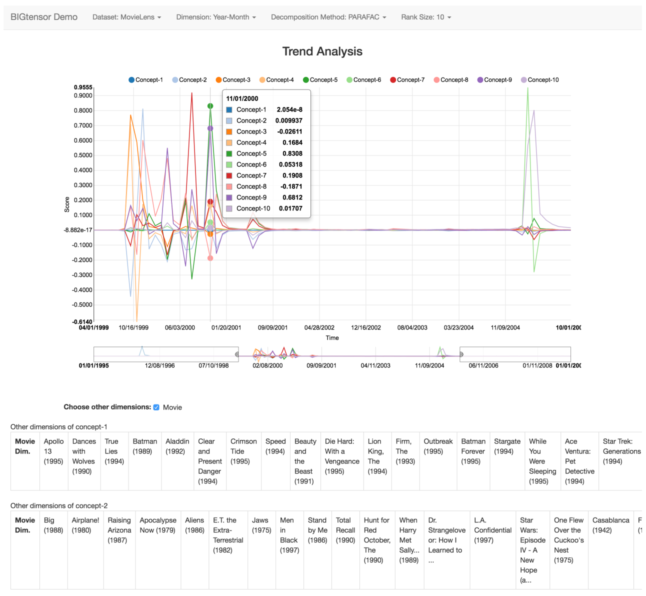
'Year-Month' Dimension

Score	Year-Month
0.77235	1999-10
0.59016	1999-11
0.20011	1999-12
0.024774	2000-07
0.024217	2000-01
0.016838	2000-02
0.014233	2000-08
0.011888	2000-12

'Movie' Dimension

Movie Id	Score	Title	Genres
1585	0.375916	Full Monty, The (1997)	Comedy, Drama
2206	0.373646	Player, The (1992)	Crime, Comedy, Drama
1714	0.311669	As Good As It Gets (1997)	Romance, Comedy, Drama
1204	0.304999	Annie Hall (1977)	Comedy, Romance
1662	0.301612	Titanic (1997)	Romance, Drama
1220	0.299279	Graduate, The (1967)	Comedy, Romance, Drama

Figure 2: Detail mode for concept discovery using PARAFAC on MovieLens dataset. The detail mode shows highly ranked elements of the selected concept with additional information such as the movie genres. It also shows a bar chart for the genre occurrences in the chosen concept to help with the discovery process of the audience. Further, the tool displays other vectors of the current concept from other factor matrices on the side.



11/01/2000	
Concept-1	2.054e-8
Concept-2	0.009937
Concept-3	-0.02611
Concept-4	0.1684
Concept-5	0.8308
Concept-6	0.05318
Concept-7	0.1908
Concept-8	-0.1871
Concept-9	0.6812
Concept-10	0.01707

Choose other dimensions: [✓ Movie]

Other dimensions of concept-1

Movie Dim.	Apollo 13 (1995)	Dances with Wolves (1990)	True Lies (1994)	Batman (1989)	Aladdin (1992)	Clear and Present Danger (1994)	Crimson Tide (1995)	Speed (1994)	Beauty and the Beast (1991)	Die Hard: With a Vengeance (1995)	Lion King, The (1994)	Firm, The (1993)	Outbreak (1995)	Batman Forever (1995)	Stargate (1994)	While You Were Sleeping (1995)	Ace Ventura: Pet Detective (1994)	Star Trek: Generations (1994)

Other dimensions of concept-2

Movie Dim.	Big (1988)	Airplane! (1980)	Raising Arizona (1987)	Apocalypse Now (1979)	Aliens (1986)	E.T. the Extra-Terrestrial (1982)	Jaws (1975)	Men in Black (1997)	Stand by Me (1986)	Total Recall (1990)	Hunt for Red October, The (1990)	When Harry Met Sally... (1989)	Dr. Strangelove or: How I Learned to ...	L.A. Confidential (1997)	Star Wars: Episode IV - A New Hope (a...	One Flew Over the Cuckoo's Nest (1975)	Casablanca (1942)	F (1

Figure 3: Trend analysis mode using PARAFAC on MovieLens dataset, in which the values of temporal elements of the selected concepts are visualized. (1) The tool supports overlapping multiple graphs. (2) The graph is zoomable; users can zoom into a shorter time frame, or zoom out to month or year level. (3) The tool displays those vectors from other factor matrices, which correspond to the concepts the user selects near the top of the tool.

B. OMITTED MATERIAL

Table 3: Summary of the tensor data used for the experiments and the demonstration. More details on each dataset can be found at http://datalab.snu.ac.kr/bigtensordemo. B: billion, M: million, K: thousand.

Name	Mode			Nonzeros
Microsoft Academic Graph	Paper	Author	Affiliation	325 M
	123 M	123 M	2.7 M	
NELL	Noun phrase 1	Noun phrase 2	Context	144 M
	26 M	26 M	48 M	
DARPA	Source IP	Dest IP	Time	28 M
	22 K	22 K	24 M	
MovieLens	User	Movie	YearMonth	10 M
	72 K	11 K	157	
YELP	User	Business	YearMonth	334 K
	71 K	16 K	108	
PhoneCall	Source	Dest	Date	1 B
	30 M	30 M	62	
Random	I	J	K	10 K~10 B
	1 K~1 B	1 K~1 B	1 K~1 B	

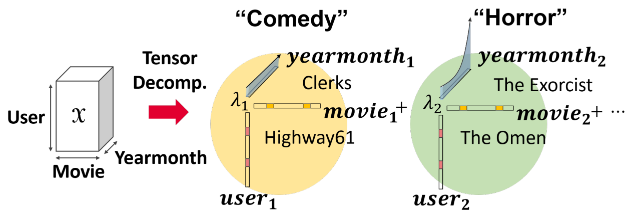


Figure 4: Overview of concept discovery and trend analysis on MovieLens dataset. We find from the movie factor matrix that movies in the same genre are clustered into the same concept, for example, comedy and horror. In the yearmonth factor matrix, we see that the demand for horror movies suddenly increases at a certain period, while the demand for comedy movies is almost constant.

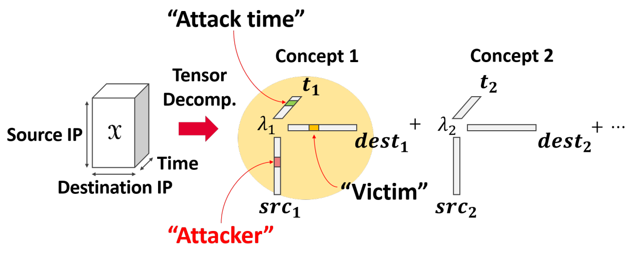


Figure 5: Overview of anomaly detection on DARPA dataset. Concept 1 shows an example of network attacks. The highest values of each vector in concept 1 indicate an attacker, a victim, and the attack time, respectively.