T-RECSYS: A Novel Music Recommendation System Using Deep Learning

Ferdos Fessahaye, Luis Perez, Tiffany Zhan, Raymond Zhang, Calais Fossier, Robyn Markarian, Carter Chiu, Justin Zhan, Laxmi Gewali, Paul Oh
University of Nevada Las Vegas, Las Vegas, United States

***Abstract*—A recommendation system is a program that utilizes techniques to suggest to a user items that they would likely prefer. This paper focuses on an approach to improving music recommendation systems, although the proposed solution could be applied to many different platforms and domains, including Youtube (videos), Netflix (movies), Amazon (shopping), etc. Current systems lack adequate efficiency once more variables are introduced. Our algorithm, Tunes Recommendation System (T-RECSYS), uses a hybrid of content-based and collaborative filtering as input to a deep learning classification model to produce an accurate recommendation system with real-time prediction. We apply our approach to data obtained from the Spotify Recsys Challenge, attaining precision scores as high as 88% at a balanced discrimination threshold.**

***Index Terms*—music recommendation; deep learning; content-based filtering; collaborative filtering;**

I. Introduction

Recommendation systems are algorithms used to make accurate suggestions based off user preferences or needs [1]. They are an often underappreciated aspect of our daily lives, influencing the way we browse videos, listening to music, and shop online. Given the wide array of applications, there are a number of decisions and approaches surrounding the design of such systems. Historically, two main approaches to recommendation existed. Content-based filtering recommends based on similarities in properties of the items [2]. If a user likes A, and B is similar to A, then a user might like B. Alternatively, collaborative filtering uses similarities between users (listening, playing, or purchasing the same/similar items) for recommendations [3]. If a user A is similar to user B, then user A might like what user B likes. Another important issue involves whether a recommender system can operate in real time, which is defined as involving pre-processing methods which are performed while data is being collected [4].

In this paper, we create a novel algorithm for recommending music to users in the form of k recommendations. Our system, Tunes Recommendation System (T-RECSYS), scores each song in a database according to user preference by utilizing a learned hybridization of content-based and collaborative filtering [5][6], returning the top-k scoring songs. The algorithm incorporates input from users utilizing historical data and preferences [7], extracting multiple key metadata variables such as genre, mood, and tempo. These variables serve as the input for our recommendation system, enabling fast and responsive recommendation. The exact mechanism for scoring ranks confidences obtained from a deep learning classification model which infers whether or not a certain song would be enjoyed by a user given the aforementioned inputs [8][9]. This approach is similar to services such as Spotify's Discover Weekly Playlist, a weekly collection of songs that Spotify recommends based on recent listening habits [10].

T-RECSYS can also be applied to the related task of playlist continuation, in which songs are recommended to the user to ”continue” a current playlist. Incidentally, this is the challenge put forth by Spotify in their RecSys Challenge 2018 [10], affirming a continuing demand for such research even while major recommendation systems are deployed on the web.

The rest of this paper is organized as follows. Section II describes the motivations behind our decision to create this algorithm, followed by Section III, which gives an overview of related work in recommendation systems. Section IV covers the methodology behind T-RECSYS. Section V illustrates and interprets experimental results obtained from an evaluation of our algorithm. Section VI discusses limitations and directions for future work, and Section VII concludes the paper.

II. Motivation

Our initial interest in recommendation systems stems from a research gap, namely the lack of multiple variable algorithms that can produce accurate recommendations with real-time updates. During our research, we found that the foundation of most algorithms for recommendation systems took into account a singular variable type [8]. The algorithms that did include more than one variable type did not include real-time updates [11]. Instead, these systems' recommendations were suggested to users based on historical data from the day before. This problem is prevalent in even well-known platforms such as Youtube and Netflix. Our interest was fueled further when we discovered the challenge issued by Spotify, the Spotify RecSys Challenge 2018, which asks teams of academic and non-academic participants to improve their already in-place recommendation system [10]. This challenge provided our dataset and allowed us to focus our broad topic of recommendation systems to specifically music recommendation systems.

III. Related Work

Articles that recognized similar problems from the lack of real-time updates or multiple variable types in recommendation systems have suggested solutions. A proposal by Fang et al. advocates a solution that first uses a questionnaire to create a user profile [11]. They state, “the information obtained from the questionnaire is compared with musical features to prune the initial 384,500 songs to 1000 songs the user is more likely to enjoy" [11]. Clustering, a class of machine learning that separates data into clusters of similar data points, is applied to the questionnaire to assign users into groups with similar musical tastes [11][12]. Through the intended use of collaborative filtering, the program combines user profiles, creating group profiles to find recommendations that are likely to be acceptable and appropriate for the whole group [11]. These recommendations focus on activities such as group physical therapies, Zumba, yoga, or pilates. Reinforcement Learning (RL) is then used to refine initial results provided by the User Profiling (UP) recommendation system [11]. The questionnaire [11] allows users to input tempo, loudness, energetic, positive, familiarity of the music, and prominence of the rhythm on a five-point scale. This allows users to adhere to strict preferences that are possibly needed during choice exercises. Although this paper includes multiple variables, collaborative, and content-based filtering, it applies specifically to exercise music, which is not suitable for everyday recommendations.

In 2017, Jiang et al. "applied a recurrent neural networks model (RNN) to make comparisons of different songs by similarity, which helps recommendation systems give ranking scores based on a combination of factors in music" [13]. RNN models are mentioned in other systems [14], but this paper measured the similarity between songs, unlike previous approaches. Although this program took into account musical factors, there was no mention of real-time updates.

Hayashi et al. proposed a fast content-based music information retrieval (CBMIR) framework called indirect matching [15]. This framework used representative queries from offline searches. They [15] integrated these offline searches as a base for providing fast similarity approximations from online sources. Similar to [13] and [16], their research included elements of content-based filtering. However, collaborative filtering and real-time updates were not elements present in the algorithm.

In 2017, Sunny et al. suggested processing real-time data streams that could provide accurate recommendations on the fly [9]. In their work, they implemented real-time updates in recommendation systems through the use of Spark and machine learning libraries, choosing to study TV channel recommendations in particular. This is similar to work from Mwinyi et al., who proposed a predictive self-learning recommendation system for multimedia content that takes into account users' preferences before and after selections are made [17]. Although both approaches implement real-time updates and multiple variable types, neither incorporated collaborative filtering into their programs. Other research from the Big Data Hub includes [18-100],

IV. ALGORITHM

The algorithm utilizes an input vector carrying information useful for both content-based and collaborative filtering. This input is fed into a deep neural network, and through a training process, the network learns to recognize patterns in a user's listening history, ultimately recommending songs it has confidence the user will enjoy.

A. Overview

T-RECSYS can be broadly described as follows. A deep learning model takes as input information representing 9 songs that are enjoyed by the listener, as well as a 10th song that represents a song that the listener may or may not like. It returns a score from 0 to 1, inclusive, that signifies the likelihood that the listener would like that 10th song based on the first 9. T-RECSYS identifies the k highest-scoring songs and recommends those to the user.

B. Content-Based Filtering

Content-based filtering in T-RECSYS is enabled by obtaining song metadata. T-RECSYS considers six categories of metadata: genre, artist type, artist era, mood, tempo, and release year. Table I illustrates an example 10-song playlist with values for each metadata field. This data can be obtained using various online APIs; in our specific experimental implementation, we accessed the Gracenote API by way of the pygn Python library. This information is instantaneous to obtain and represents a lightweight encoding of the content of a song. When provided as input into the deep neural network as described below, a content-based filtering approach can be learned by the model.

Of the six metadata categories, three are unorderable (genre, artist type, and mood) while the remaining three (artist era, tempo, and release year) are orderable. Therefore, for the purposes of the input vector for the deep learning model (which only accepts numeric values), the unorderable data are one-hot encoded – a machine learning technique where a categorical variable is converted into multiple boolean variables. Table II illustrates an example of one-hot encoding for genre, artist type, and mood, demonstrating that there is a unique one-hot encoding for each metadata class.

C. Collaborative Filtering

Collaborative filtering is also present in T-RECSYS. In particular, it manifests itself as a measure of the pair similarity between each of the nine songs enjoyed by the user and the tenth song. That similarity is defined as the frequency with which the two songs appear together in a dataset of playlists. Intuitively, if the songs frequently appear in playlists together, one is a likely recommendation for the other. T-RECSYS incorporates this similarity score in two slightly different ways. One is a volume measure, in which simply the total number of playlists where each pair of songs appear

TABLE I: Example playlist when defined by song metadata

Playlist 1	**Song 1**	**Song 2**	**Song 3**	**Song 4**	**Song 5**	**Song 6**	**Song 7**	**Song 8**	**Song 9**	**Song 10**
Genre	Urban	Rock	Country	Pop	Rap	Pop	Classical	Hip-Hop	Pop	Country
Artist Type	Female Group	Female Solo	Male Duo	Female Solo	Male Duo	Male Solo	Male Duo	Female Duo	Male Group	Female Group
Artist Era	2000's	1950's	1930's	2010's	1940's	1990's	1970's	1930's	1960's	1990's
Mood	Happy	Angry	Sad	Party	Sad	Energetic	Dance	Sad	Energetic	Happy
Tempo	Fast	Fast	Medium	Slow	Medium	Fast	Medium	Medium	Medium	Fast
Release Year	2001	1962	1930	2009	1949	1986	1977	1932	1966	2010

TABLE II: Example of one-hot encoding for different song metadata

Genre	Binary	Artist Type	Binary	Mood	Binary
Hip-Hop	0000001000	Female Duos	0000100000	Sad	0010000000
Country	0010000000	Male Duos	0010000000	Sad	0010000000
Pop	1000000000	Female Group	1000000000	Happy	1000000000
Rap	0000100000	Male Group	0001000000	Sad	0010000000
Rock	0100000000	Female Solo	0100000000	Angry	0100000000
Urban	0001000000	Female Solo	0100000000	Party	0001000000

is recorded. However, this measure does not take into account how often each song appears independently. Two songs may appear together many times, but perhaps they appear without each other far more often. To accomodate for this, T-RECSYS also measures the Sorenson index for each pair:

$$\frac{2|X \cap Y|}{|X| + |Y|},$$

where X and Y represent the set of playlists that the first song and second song in the pair appear in respectively. Altogether, two measures for nine song pairs results in 18 values representing the portion of the input vector dedicated to collaborative filtering.

D. Deep Learning Model

The input vector is assembled by combining the content-based and collaborative filtering information as described above. The intention is for this data to be used first to train a deep neural network to learn the concept of song recommendation, and then to actually give accurate song recommendations.

Experimentation with different hyperparameter configurations resulted in the network structure specification described in Table III. We utilized Google's Tensorflow together with the Python deep learning library Keras to build, train, and deploy our model. For the training process, we modified the playlist dataset provided by Spotify [10] to obtain a training set. Each playlist containing ten or more songs was used to build two training instances. In the first instance, the first nine songs in the playlist were used to evaluate whether or not the user would like the tenth, and corresponded to a positive recommendation (since the user who built the playlist liked all ten songs). In the second instance, the first nine songs were used to evaluate whether or not the user would like a random song, and corresponded to a negative recommendation. There is no way of knowing whether the user would actually like the randomly selected song, but probability favors the negative case. Once all training instances were built in this manner, 20% of the training set was separated at random to construct a testing set for evaluation purposes.

TABLE III: Neural network configuration

Layer	**Parameters**
Fully Connected	Nodes: 70, Activation: ReLU
Dropout	Dropout: 80%
Fully Connected	Nodes: 20, Activation: ReLU
Dropout	Dropout: 40%
Fully Connected	Nodes: 7, Activation: ReLU
Dropout	Dropout: 20%
Fully Connected	Nodes: 3, Activation: ReLU
Output	Nodes: 1, Activation: Sigmoid

V. Results

To evaluate the model, we chose to use precision as the chief metric, defined as

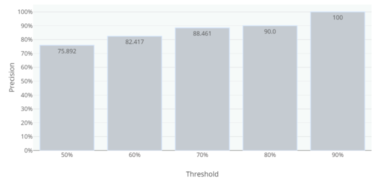
$$precision = \frac{t_p}{t_p + f_p},$$

where t_p indicates true positives and f_p indicates false positives. Specifically in the context of song recommendations, precision can be described as how often a song that was recommended was actually enjoyed. This aligns with our intuitive notion of how recommendation systems should work. Another configurable variable of note is the discrimination threshold. The discrimination threshold determines how high of a confidence score (as predicted by the deep neural network) should be obtained before a recommendation is made. By default, a recommendation might be made when the confidence score exceeds 50% since it is predicted that the listener would more likely than not like the song, but the threshold can be raised so that only songs with high confidence (i.e., high probability of being liked) are recommended.

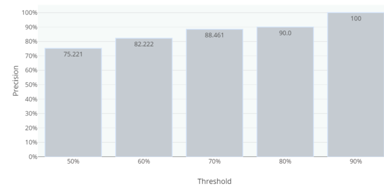
Figures 1a to 1d illustrate 4 separate test trials, depicting precision obtained at various discrimination thresholds. As can be observed, by utilizing a 90% threshold, perfect precision was achieved in 3 out of 4 trials, meaning every single song that was recommended was enjoyed. Even when using the default 50% threshold, impressive precision scores can be obtained, with Trial 3 scoring over 88% precision. The sample standard deviation between the 4 trials was 8.58%, indicating that the procedure is fairly consistent.

VI. Discussion

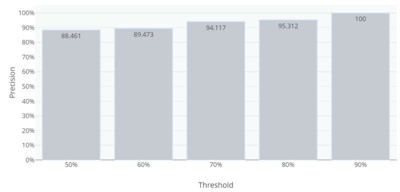
Overall, we were certainly impressed with the precision scores we were able to achieve. There are a few limitations with our setup worth noting. One concern was the lack of ground truth on negative recommendations. Utilizing solely a collection of playlists, there is no way of knowing the songs that a user would not like. However, this is a dataset limitation rather than a inherent issue with T-RECSYS as this could be



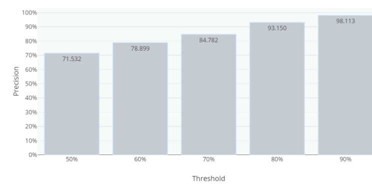
(a) Trial 1



(b) Trial 2



(c) Trial 3



(d) Trial 4

Fig. 1: Precision vs. Discrimination Threshold

easily remedied with user data about disliked songs, which is most certainly collected by Spotify and any other music service. Another drawback with the approach is the way that collaborative filtering data is obtained. Calculating both values requires global information about every song and playlist, which can be expensive to maintain. However, if deployed at a large scale, the utilization of industry-standard indexing techniques should make this drawback negligible.

Avenues for future research are clear to see. Within the domain of music recommendation, more songs (rather than the 9 used in this paper) could be used to establish the user preferences baseline. For each song, more types of metadata could be used, or even intrinsic musical features of the song itself perhaps. Parallelization via Spark could be performed to increase training and recommendation efficiency. Extensions of the T-RECSYS framework to videos, movies, shopping, and any other domain could be performed as well. Like songs, each item has numerically representable features that can be encoded as described in the paper.

VII. Conclusion

Recommendation systems are an overlooked part of our daily lives that frequently dictate the music we listen to on Spotify and the videos we watch on YouTube. Research continues to advance the sophistication and precision of these systems. In our paper, we describe one such recommendation system, T-RECSYS, that takes into account both content-based and collaborative filtering as input into a deep learning model that learns user music preferences to create song recommendations, similar to the systems employed by Spotify, Pandora, and iTunes. In the development of our algorithm, we sought to overcome recurring problems in existing algorithms from the literature, such as the lack of real-time updates and multiple variable input types. The result is a system capable of achieving high recommendation precision and readily extensible to different market services such as Amazon or Netflix.

Acknowledgment

Funding for this research was provided by the Army Educational Outreach Program (AEOP) and the National Science Foundation's Research Experiences for Teachers (RET) program. We also want to acknowledge the UNLV College of Engineering, Writing Center and Department of Communication Studies for their support.

References

[1] P. Jomsri, S. Sanguansintukul, and W. Choochaiwattana, "A framework for tag-based research paper recommender system: An ir approach," in *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, April 2010, pp. 103–108.

[2] M. Hassan and M. Hamada, "Improving prediction accuracy of multi-criteria recommender systems using adaptive genetic algorithms," in *2017 Intelligent Systems Conference (IntelliSys)*, Sept 2017, pp. 326–330.

[3] Ruchika, A. V. Singh, and M. Sharma, "Building an effective recommender system using machine learning based framework," in *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, Dec 2017, pp. 215–219.

[4] Y. Im, P. Prahladan, T. H. Kim, Y. G. Hong, and S. Ha, "Snn-cache: A practical machine learning-based caching system utilizing the inter-relationships of requests," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, March 2018, pp. 1–6.

[5] L. Shou-Qiang, Q. Ming, and X. Qing-Zhen, "Research and design of hybrid collaborative filtering algorithm scalability reform based on genetic algorithm optimization," in *2016 6th International Conference on Digital Home (ICDH)*, Dec 2016, pp. 175–179.

[6] I. Naser, R. Pagare, N. Wathap, and V. Pingale, "Hybrid music recommendation system: Enhanced collaborative filtering using context and interest based approach," in *2014 Annual IEEE India Conference (INDICON)*, Dec 2014, pp. 1–11.

[7] J. Lee, S. Shin, D. Jang, S. J. Jang, and K. Yoon, "Music recommendation system based on usage history and automatic genre classification," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2015, pp. 134–135.

[8] Y. Feng, Y. Zhuang, and Y. Pan, "Music information retrieval by detecting mood via computational media aesthetics," in *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, Oct 2003, pp. 235–241.

[9] B. K. Sunny, P. S. Janardhanan, A. B. Francis, and R. Murali, "Implementation of a self-adaptive real time recommendation system using spark machine learning libraries," in *2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, Aug 2017, pp. 1–7.

[10] C.-W. Chen, C. D. Boom, J. Garcia-Gathright, P. Lamere, J. McInerney, V. Murali, H. Rawlinson, S. Reddy, and R. Yon, "The million playlist dataset," Spotify - RecSys Challenge 2018, 2018. [Online]. Available: https://recsys-challenge.spotify.com/dataset

[11] J. Fang, D. Grunberg, S. Luit, and Y. Wang, "Development of a music recommendation system for motivating exercise," in *2017 International Conference on Orange Technologies (ICOT)*, Dec 2017, pp. 83–86.

[12] M. Ahmed, M. T. Imtiaz, and R. Khan, "Movie recommendation system using clustering and pattern recognition network," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2018, pp. 143–147.

[13] M. Jiang, Z. Yang, and C. Zhao, "What to play next? a rnn-based music recommendation system," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct 2017, pp. 356–358.

[14] M. Kataoka, M. Kinouchi, and M. Hagiwara, "Music information retrieval system using complex-valued recurrent neural networks," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, vol. 5, Oct 1998, pp. 4290–4295 vol.5.

[15] T. Hayashi, N. Ishii, M. Ishimori, and K. Abe, "Stability improvement of indirect matching for music information retrieval," in *2015 IEEE International Symposium on Multimedia (ISM)*, Dec 2015, pp. 229–232.

[16] X. Wang, Y. Bai, and Y. Li, "An information retrieval method based on sequential access patterns," in *2010 Asia-Pacific Conference on Wearable Computing Systems*, April 2010, pp. 247–250.

[17] I. H. Mwinyi, H. S. Narman, K. C. Fang, and W. S. Yoo, "Predictive self-learning content recommendation system for multimedia contents," in *2018 Wireless Telecommunications Symposium (WTS)*, April 2018, pp. 1–6.

[18] C. Chiu and J. Zhan, "Deep learning for link prediction in dynamic networks using weak estimators," *IEEE Access*, vol. 6, no. 1, pp. 35 937 – 35 945, 2018.

[19] M. Bhaduri and J. Zhan, "Using empirical recurrences rates ratio for time series data similarity," *IEEE Access*, vol. 6, no. 1, pp. 30 855–30 864, 2018.

[20] J. Wu, J. Zhan, and S. Chobe, "Mining association rules for low frequency itemsets," *PLOS ONE*, vol. 13, no. 7, 2018.

[21] P. Ezatpoor, J. Zhan, J. Wu, and C. Chiu, "Finding top-k dominance on incomplete big data using mapreduce framework," *IEEE Access*, vol. 6, no. 1, pp. 7872–7887, 2018.

[22] P. Chopade and J. Zhan, "Towards a framework for community detection in large networks using game-theoretic modeling," *IEEE Transactions on Big Data*, vol. 3, no. 3, pp. 276–288, 2017.

[23] M. Bhaduri, J. Zhan, and C. Chiu, "A weak estimator for dynamic systems," *IEEE Transactions on Big Data*, vol. 5, no. 1, pp. 27 354–27 365, 2017.

[24] M. Pirouz and J. Zhan, "Toward efficient hub-less real time personalized pagerank," *IEEE Transactions on Big Data*, vol. 5, no. 1, pp. 26 364–26 375, 2017.

[25] M. Bhaduri, J. Zhan, C. Chiu, and F. Zhan, "A novel online and non-parametric approach for drift detection in big data," *IEEE Access*, vol. 5, no. 1, pp. 15 883–15 892, 2017.

[26] C. Chiu, J. Zhan, and F. Zhan, "Uncovering suspicious activity from partially paired and incomplete multimodal data," *IEEE Access*, vol. 5, no. 1, pp. 13 689 – 13 698, 2017.

[27] R. Ahn and J. Zhan, "Using proxies for node immunization identification on large graphs," *IEEE Access*, vol. 5, no. 1, pp. 13 046–13 053, 2017.

[28] M. Wu, J. Zhan, and J. Lin, "Ant colony system sanitization approach to hiding sensitive itemsets," *IEEE Access*, vol. 5, no. 1, pp. 10 024–10 039, 2017.

[29] J. Zhan and B. Dahal, "Using deep learning for short text understanding," *Journal of Big Data*, vol. 4, no. 34, pp. 1–15, 2017.

[30] J. Zhan, S. Gurung, and S. P. K. Parsa, "Identification of top-k nodes in large networks using katz centrality," *Journal of Big Data*, vol. 4, no. 16, 2017.

[31] J. Zhan, T. Rafalski, G. Stashkevich, and E. Verenich, "Vaccination allocation in large dynamic networks," *Journal of Big Data*, vol. 4, no. 2, pp. 161–172, 2017.

[32] J. M.-T. Wu, J. Zhan, and J. C.-W. Lin, "An aco-based approach to mine high-utility itemsets," *Knowledge-Based Systems*, vol. 116, pp. 102–113, 2017.

[33] M. Pirouz, J. Zhan, and S. Tayeb, "An optimized approach for community detection and ranking," *Journal of Big Data*, vol. 3, no. 22, pp. 102–113, 2017.

[34] J. Zhan, V. Gudibande, and S. P. K. Parsa, "Idenfication of top-k influential communities in large networks," *Journal of Big Data*, vol. 3, no. 16, 2016.

[35] M. Pirouz and J. Zhan, "Node reduction in personalized page rank estimation for large graphs," *Journal of Big Data*, vol. 3, no. 12, 2016.

[36] H. Selim and J. Zhan, "Towards shortest path identification on large networks," *Journal of Big Data*, vol. 3, no. 10, 2016.

[37] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 5, pp. 1–14, 2015.

[38] P. Chopade and J. Zhan, "Structural and functional analytics for community detection in large-scale complex networks," *Journal of Big Data*, vol. 2, no. 1, pp. 1–28, 2015.

[39] J. Zhan and X. Fang, "A computational framework for detecting malicious actors in communities," *International Journal of Privacy, Security, and Integrity*, vol. 2, no. 1, pp. 1–20, 2014.

[40] A. Rajasekar, H. Kum, M. Cross, J. Crabtree, S. Sankaran, H. Lander, T. Carsey, G. King, and J. Zhan, "The databridge," *Science Journal*, vol. 2, no. 1, pp. 1–14, 2013.

[41] J. Zhan, X. Fang, and N. Koceja, "A novel framework on data reduction," *Science Journal*, vol. 2, no. 1, pp. 15–23, 2013.

[42] A. Doyal and J. Zhan, "Towards ddos defense and traceback," *International Journal of Privacy, Security, and Integrity*, vol. 1, no. 4, pp. 299–311, 2013.

[43] J. Zhan and X. Fang, "Towards social network evolution," *Human Journal*, vol. 1, no. 1, pp. 218–233, 2012.

[44] J. Zhan, J. Oommen, and J. Crisostomo, "Anomaly detection in dynamic systems using weak estimator," *ACM Transaction on Internet Technology*, vol. 11, no. 1, pp. 53–69, 2011.

[45] J. Zhan and X. Fang, "Social computing: The state of the art," *International Journal of Social Computing and Cyber-Physical Systems*, vol. 1, no. 1, pp. 1–12, 2011.

[46] N. Mead, M. S., and J. Zhan, "Integrating privacy requirements considerations into a security requirements engineering method and tool," *International Journal of Information Privacy, Security and Integrity*, vol. 1, no. 1, pp. 106–126, 2011.

[47] J. Zhan, "Granular computing in privacy-preserving data mining," *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, vol. 1, no. 3, pp. 272–288, 2010.

[48] J. Wang, J. Zhang, and J. Zhan, "Towards real-time performance of data privacy protection," *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, vol. 1, no. 4, pp. 329–342, 2010.

[49] J. Zhan, "Secure collaborative social networks," *IEEE Transaction on Systems, Man, and Cybernetics, Part C*, vol. 40, no. 6, pp. 682–689, 2010.

[50] J. Zhan, H. C., I. Wang, T. Hsu, C. Liau, and W. D., "Privacy-preserving collaborative recommender systems," *IEEE Transaction on Systems, Man, and Cybernetics, Part C*, vol. 40, no. 4, pp. 472–476, 2010.

[51] H. Park, J. Hong, J. Park, J. Zhan, and D. Lee, "Attribute-based access control using combined authentication technologies," *IEEE Transaction on Mobile Computing*, vol. 9, no. 6, pp. 824–837, 2010.

[52] I. Wang, C. Shen, J. Zhan, T. Hsu, C. Liau, and D. Wang, "Empirical evaluations of secure scalar product," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 39, no. 4, pp. 440–447, 2009.

[53] A. Inoue, T. Wong, and J. Zhan, "Applications of machine learning to information security and privacy," *Journal of Japanese Society for Fuzzy Theory and Intelligent Informatics*, vol. 19, no. 3, pp. 222–232, 2009.

[54] J. Zhan, "Privacy-preserving collaborative data mining," *IEEE Computational Intelligence Magazine*, vol. 3, no. 2, pp. 31–41, 2008.

[55] A. Bashir and J. Zhan, "Not always a blunt tool legislation in the context of privacy externalities," *Communications of the Chinese Cryptology and Information Security Association*, vol. 2, no. 1, pp. 36–48, 2008.

[56] J. Zhan and V. Rajamani, "The economic aspects of privacy," *International Journal of Security and Its Applications*, vol. 2, no. 3, pp. 101–108, 2008.

[57] J. Zhan, "The economic aspects of privacy," *International Journal of Security and Its Applications*, vol. 2, no. 3, pp. 101–108, 2008.

[58] W. Zhang, P. Wang, K. Peace, J. Zhan, and Y. Zhang, "On truth, uncertainty, and bipolar logic," *Journal of New Mathematics and Natural Computing*, vol. 4, no. 2, pp. 55–65, 2008.

[59] N. Mead, V. Viswanathan, and J. Zhan, "Incorporating security requirements engineering into standard lifecycle processes," *International Journal of Security and Its Applications*, vol. 2, no. 4, pp. 67–80, 2008.

[60] J. Zhan and S. Matwin, "Privacy-preserving data mining in electronic surveys," *International Journal of Network Security*, vol. 4, no. 3, pp. 318–327, 2007.

[61] J. Zhan, L. Chang, and S. Matwin, "Privacy-preserving multi-party decision tree induction," *International Journal of Business Intelligence and Data Mining*, vol. 2, no. 2, pp. 197–212, 2007.

[62] J. Zhan, S. Matwin, and L. Chang, "Privacy-preserving collaborative association rule mining," *Journal of Network and Computer Applications*, vol. 30, no. *, pp. 1216–1227, 2007.

[63] J. Zhan, L. Chang, and S. Matwin, "Building k-nearest neighbor classifiers on vertically partitioned private data," *International Journal of Network Security*, vol. 1, no. 1, pp. 46–51, 2005.

[64] J. Zhan and S. Matwin, "Privacy preserving support vector machine classification," *International Journal of Intelligent Information and Database Systems*, vol. 1, no. 3/4, pp. 356–385, 2005.

[65] S. Matwin, L. Chang, R. Wright, and J. Zhan, "Editorial - privacy and security aspects of data mining special issue," *International Journal of Information and Computer Security*, vol. 2, no. 1, p. 1, 2005.

[66] J. Zhan, S. Matwin, and L. Chang, "Privacy-preserving electronic voting," *International Journal of Information and Security*, vol. 15, no. 2, pp. 165–180, 2004.

[67] L. Singh and J. Zhan, "Measuring topological anonymity in social networks," in *Proceedings of the IEEE International Conference on Granular Computing*, Silicon Valley, USA, November 2007, pp. 770–774.

[68] C. Barron, H. Yu, and J. Zhan, "Cloud computing security case studies and research," in *2013 International Conference of Parallel and Distributed Computing*, London, UK, July 2013.

[69] H. Park, B. Kim, D. Lee, Y. Chung, and J. Zhan, "Secure similarity search," in *Proceedings of the IEEE International Conference on Granular Computing*, Silicon Valley, USA, November 2007, pp. 598–598.

[70] S. Miyazaki, N. Mead, and J. Zhan, "Computer-aided privacy requirements elicitation techniques," in *Proceedings of the IEEE Asia Pacific International Conference on Services Computing*, Yilan, Taiwan, December 2008, pp. 367–372.

[71] X. Fang and J. Zhan, "Online banking authentication using mobile phones," in *Proceedings of the International Symposium on Financial Security*. Busan, Korea: IEEE CS Press, May 2010.

[72] J. Zhan and L. Chang, "Privacy-preserving collaborative sequential pattern mining with horizontally partitioned datasets," in *Proceedings of the International Conference on Data Privacy and Security in A Global Society*, Skiathos, Greece, May 2004, pp. 242–252.

[73] T. Yu, D. Lee, and J. Zhan, "Multi-party k-means clustering with privacy consideration," in *Proceedings of IEEE International Symposium on Parallel and Distributed Processing with Applications*, Taipei, Taiwan, September 2010.

[74] Y. Duan, J. Canny, and J. Zhan, "Efficient privacy-preserving association rule mining: P4p style," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, Honolulu, Hawaii, USA, April 2007, pp. 654–660.

[75] J. Zhan and X. Fang, "A novel trust computing system for social networks," in *IEEE International Conference on Privacy, Security, Risk and Trust*. MIT, Boston, USA: IEEE CS Press, October 2011.

[76] J. Zhan and V. Rajamani, "The economics of privacy: People, policy and technology," in *Proceedings of the International Conference on Information Security and Assurance*. IEEE CS Press, April 2008, pp. 579–584.

[77] G. Blosser and J. Zhan, "Privacy-preserving collaborative social networks," in *Proceedings of the International Conference on Information Security and Assurance*. IEEE CS Press, April 2008, pp. 543–548.

[78] K. Prakobphol and J. Zhan, "A novel outlier detection scheme for network intrusion detection systems," in *Proceedings of the International Conference on Information Security and Assurance*. IEEE CS Press, April 2008, pp. 555–560.

[79] C. Shen, J. Zhan, D. Wang, T. Hsu, and C. Liau, "Information theoretically secure number product protocol," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, Hong Kong, August 2007, pp. 3006–3011.

[80] C. Hsieh, J. Zhan, D. Zeng, and F. Wang, "Privacy-preserving in joining recommender systems," in *Proceedings of the International Conference on Information Security and Assurance*. IEEE CS Press, April 2008, pp. 561–566.

[81] J. Zhan, G. Blosser, C. Yang, and L. Singh, "Privacy-preserving collaborative social networks," in *Lecture Notes in Computer Science*, vol. 5075. Springer, 2008, pp. 114–125.

[82] J. Zhan and X. Fang, "Trust maximization in social networks," in *Proceedings of International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, Maryland, USA, March 2011.

[83] ——, "A computational trust framework for social computing," in *Proceedings of IEEE International Conference on Social Computing*. Minneapolis, USA: IEEE CS Press, August 2010, pp. 1–5.

[84] J. Zhan and L. Thomas, "Phishing detection using stochastic learning-based weak estimators," in *IEEE Symposium Series on Computational Intelligence*. Paris, France: IEEE, April 2011.

[85] J. Zhan and X. Fang, "Authentication using multi-level social networks," in *International Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Springer Series on Communications in Computer and Information Science*, vol. 128, 2011.

[86] J. Zhan, I. Cheng, C. Hsieh, T. Hsu, C. Liau, and D. Wang, "Towards efficient privacy-preserving collaborative recommender systems," in *Proceedings of the IEEE International Conference on Granular Computing*, Hangzhou, China, August 2008, pp. 778–783.

[87] C. Hsu, J. Zhan, W. Fang, and J. Ma, "Towards improving qos-guided scheduling in grids," in *Proceedings of the 3rd ChinaGrid Annual Conference*, Dunhuang,China, August 2008, pp. 89–95.

[88] X. Fang, N. Koceja, J. Zhan, G. Dozier, and D. Dasgupta, "An artificial immune system for phishing detection," in *IEEE Congress on Evolutionary Computation*. Brisbane, Australia: IEEE Press, June 2012.

[89] H. Park, D. Lee, J. Zhan, and G. Blosser, "Efficient keyword index search over encrypted data of groups," in *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, June 2008, pp. 225–229.

[90] H. Park, J. Zhan, and D. Lee, "Privacy-aware access control through negotiation in daily life service," in *International Conference on Intelligence and Security*, vol. 5075. Springer, 2008, pp. 514–519.

[91] J. Zhan, S. Matwin, and L. Chang, "Private mining of association rules," in *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, Atlanta, Georgia, May 2005, pp. 72–80.

[92] C. Shen, J. Zhan, T. Hsu, C. Liau, and D. Wang, "Scalar product-based secure two party computation," in *Proceedings of the IEEE International Conference on Granular Computing*, Hangzhou, China, August 2008, pp. 600–605.

[93] X. Xu, J. Zhan, and H. Zhu, "Using social networks to organize researcher community," in *International Conference on Intelligence and Security Informatics*, vol. 5075. Springer, 2008, pp. 421–427.

[94] G. Blosser and J. Zhan, "Privacy-preserving collaborative e-voting," in *International Conference on Intelligence and Security Informatics*, vol. 5075. Springer, 2008, pp. 508–513.

[95] J. Zhan, L. Chang, and S. Matwin, "Privacy-preserving multi-party decision tree classification," in *Proceedings of the 2004 Annual IFIP WG 11.3 Working Conference on Data and Application Security*, Sitges, Catalonia, Spain, July 2004, pp. 341–355.

[96] ——, "Building k-nearest neighbor classification on vertically partitioned private data," in *Proceedings of the IEEE International Conference on Granular Computing*, Beijing, China, July 2005, pp. 708–711.

[97] P. Chopade and J. Zhan, "Community detection in large scale big data networks," in *The Second ASE International Conference on Big Data*, Stanford, CA, USA, May 2014.

[98] J. Wang, J. Zhan, and J. Zhang, "Towards real-time performance of data value hiding for frequent data updates," in *Proceedings of the IEEE International Conference on Granular Computing*, Hangzhou, China, August 2008, pp. 606–611.

[99] J. Zhan, L. Chang, and S. Matwin, "Privacy-preserving naive bayesian classification over horizontally partitioned data," in *Data Mining: Foundations and Practice, Edited by T.Y. Lin and Y. Xie*. Hong Kong: Springer-Verlag, November 2006, pp. 529–538.

[100] P. Chopade, J. Zhan, K. Roy, and K. Flurchick, "Real-time large-scale big data networks analytics and visualization architecture," in *The 12th International Conference on Emerging Technologies for a Smarter World*, New York, USA, October 2015.