Context Data Compact Prediction Tree (CD-CPT): Transforming User Experience Through Predictive Analysis

Pooja Goyal[1], Md Khorrom Khan[1], Natnael Teshome[1], Brendan Geary[2] and Renee Bryce[1]
[1] *Computer Science & Engineering, University of North Texas, Denton, Texas, U.S.A.*
[2] *Computer Science, Florida Polytechnic University, Florida, U.S.A.*

Keywords: Context-Aware Applications, Sequence Prediction, Sequential Rule Mining, Compact Prediction Tree, Transition Directed Acyclic Graph, Prediction by Pattern Mining, All-k Order Markov, Dependency Graph, Android Testing, Context Aware Environments, Mobile Application Testing.

Abstract: Use of IoT (Internet of Things) devices have significantly increased over the last decade, specifically smartphones as compared to desktops, and laptops have become an integral part of our everyday lives. Smartphone applications operate in dynamic environments and generate huge and vast amount of context events such as screen orientation, location, battery life, and network connectivity throughout the day. Such context events may affect usage of the smartphone and smartphone applications by the user and the behaviour of these applications, Sparsity and complexity of these events make it difficult to identify patterns and trends in the data using traditional data mining techniques. Hence, predictive analysis of these events and finding patterns in context event data can have drastic impact on the application usage and enhance user experience. Prediction trees can be used to predict future events based on the context of past events, This work proposes a modified method of Compact Prediction Tree (CPT) called Context Data Compact Prediction Tree (CD-CPT) to predict real-world context data for multiple users. The experiments conducted used Transition Directed Acyclic Graph (TDAG) and All-k Order Markov (AKOM) algorithms to generate short-term predictions based on current context events and compare with baseline models such as Prediction by Pattern Mining (PPM), Dependency Graph (DG), CPT, and CPT+. The experimental results indicate that AKOM and TDAG outperform other algorithms, achieving a 50.4% weighted F-1 score for the highest supported context event. CD-CPT, without referencing the test file, still achieves a 14.27% weighted F-1 score for the same event, showing potential for improved accuracy in predicting context data compared to other algorithm.

1 INTRODUCTION

As smartphone usage continues to rise worldwide, the growing availability of these devices necessitates the optimization of applications for improved efficiency and security (Data.ai, 2022).Numerous devices employ context-aware apps that adapt to alterations in their surroundings as android applications are evolving to be more complex and sophisticated (and, 2022). Smartphones, equipped with complex hardware and software, generate an extensive range of data about context events, i.e., connecting to WiFi, connecting/disconnecting a headset, changing screen orientation, bluetooth, location changes, etc (Rahmati and Zhong, 2012). By interpreting this data, we can better utilize time and resources, while also providing developers with valuable information for testing applications (Goyal et al., 2023) with respect to important sequences of context events.

This study primarily aims to determine if algorithms from SPMF (Fournier Viger, 2016), including a modified version of the Compact Prediction Tree (CPT) (Gueniche et al., 2013) called Context Data Compact Prediction Tree (CD-CPT), Transition Directed Acyclic Graph (TDAG)(Laird and Saul, 1994) and All-k Order Markov (AKOM)(Pitkow and Pirolli, 1999), may identify patterns in real-world context data sequences from users. We compare AKOM and TDAG to a majority baseline of algorithms, including CPT, Compact Prediction Tree+ (CPT+)(Gueniche et al., 2015), Dependency Graph (DG)(Padmanabhan and Mogul, 1996), and Prediction by Partial Matching (PPM)(Cleary and Witten, 1984). We utilize a dataset of context data from 58 real-world Android users on different devices for 30 day periods using the ContextMon application(Piparia et al., 2021)

Goyal, P., Khan, M., Teshome, N., Geary, B. and Bryce, R.
Context Data Compact Prediction Tree (CD-CPT): Transforming User Experience Through Predictive Analysis.
DOI: 10.5220/0012615800003705

In *Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS 2024)*, pages 166-173
ISBN: 978-989-758-699-6; ISSN: 2184-4976

(Goyal et al., 2023). We modified the Java version of SPMF(Fournier-Viger et al., 2014) to enable the CPT algorithm to generate multiple context data sequences representing different use cases.

This work uses SPMF (Fournier Viger, 2016) i.e, a java based library that have around 250+ algorithms, this work modifies SPMF's CPT model to make relevant predictions, referring to this modified version as CD-CPT. In order to train the CD-CPT model, we utilized real-world context data sequences from 15 Android devices, as the CPT model portion of CD-CPT had limitations and we could not use all 47 sequences for training. Subsequently, we employed this trained model to predict context event sequences for the next 11 Android devices, and compared these results with the sequences of context data obtained from the remaining 11 real-world Android devices volunteered by users. As for AKOM, TDAG, PPM, CPT+, CPT, and DG models, we utilized all 47 sequences as a train file, as it did not take much time to train these models. For predicting the next context event, we utilized a window of two for PPM and DG, a window of five for CPT and CPT+, and a window of four for AKOM and TDAG. These different window sizes proved to be beneficial in creating the best performing parameters for each of the algorithms.

The following sections in this paper are arranged as follows: Section II describes background information on context events and related concepts, Section III explores the use of CD-CPT, AKOM, TDAG, PPM, CPT, CPT+, and DG models. Section IV covers data collection, algorithm model implementation, and the modification of the Java version of SPMF's CPT model to develop CD-CPT. Section V outlines the evaluation metrics and experimental setup. Section VI assesses the prediction success of CD-CPT, TDAG, and AKOM, and discusses the results derived from the study. Finally, Section VII of the paper provides the conclusion and outlines potential avenues for future exploration.

2 BACKGROUND AND RELATED WORK

In this section, we provide a review of background work in context-aware computing, context event prediction, and optimization of smartphone applications, and highlight how our research differs from previous works in these areas. To better understand the significance and relevance of this research, we delve into the broader field of context-aware computing, context event prediction, and optimization of smartphone applications.

Context events in smartphone applications are a 2-tuple (x, y), where *x* denotes the context category and *y* represents the context action. These events may influence the way a smartphone application reacts(Dey, 2001), By incorporating context-awareness in the algorithms for predicting context events, our paper aims to improve the efficiency and personalization of smartphone applications. Examples of context events includes changes to network connections, volume adjustments, battery level changes, and more. Some applications may adjust behavior due to a context event change. For instance, an app may choose to respond differently when the battery is low. Research on context events in smartphones over the past few years(Rahmati and Zhong, 2012) have highlighted the potential applications and limitations of this knowledge. By understanding context events, developers can optimize applications for improved user experience and energy efficiency. However, challenges arise due to the complexity of context events and concerns about user privacy.Within the field of smartphone testing for context events, a variety of approaches such as sensor-based testing, user-based testing, and hybrid testing are typically used to achieve a thorough insight into context usage. Despite this, these techniques struggle with achieving accuracy, managing time complexity, and representing a wide range of users in the simulation of real-world events (Bosmans et al., 2019). Sequence prediction strives to forecast the next event or symbol in a sequence based on historical data. Several sequence prediction algorithms exist, addressing different aspects of sequence prediction and offering diverse levels of performance, complexity, and applicability. Prediction by Partial Matching (PPM)(Cleary and Witten, 1984), a fast and simple sequence prediction model that remains popular today, despite being less accurate than some newer models. PPM has been used in various applications, such as identifying manufacturing patterns.

An Another study i.e, AppsPred (Sarker and Salah, 2019) is a data-driven model that utilizes real-world data collected from university students to predict smartphone app usage based on daily life activities. This model's performance is attributed to its optimal use of decision trees within a forest, outperforming other machine learning techniques. However, the study's dataset was limited in size and only focused on single-user predictions. In contrast, our CD CPT model predicts event sequences by analyzing data from multiple users. In addition, the "BehavDT"(Sarker et al., 2020) model addressed the problem of building behavioral activities using a context-aware predictive model by considering individual user preference levels. It is worth noting that

CD CPT and BehavDT differ in their primary focus; while our model predicts event sequences, BehavDT is designed to build behavioral activities using a context-aware predictive model.

The Dependency Graph (DG)(Padmanabhan and Mogul, 1996) algorithm, a sequence prediction model that utilizes directed graphs to describe dependencies between entities in a system. DG has demonstrated good performance and memory efficiency in different applications. The Compact Prediction Tree (CPT)(Gueniche et al., 2013) algorithm, a novel sequence prediction model that losslessly compresses training data with low time complexity. The model has been applied to robotic systems for predicting event sequences and enabling quick learning. Building on the CPT model, proposed CPT+ (Gueniche et al., 2015), an improved version designed to reduce time and space complexity. CPT+ has shown significant improvements in performance and efficiency compared to the original CPT.

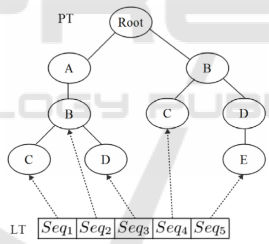
The Compact Prediction Tree (CPT)(Gueniche et al., 2013) is a sequence prediction algorithm chosen for this study due to its ability to losslessly compress training data, ensuring all relevant information is retained for subsequent predictions (Mani and Suneetha, 2020). However, the original CPT model outputs data for only one predicted context event rather than an entire sequence of predictions for a new user's context data.To address this limitation, we implemented our own modifications to CPT, creating the Context Data Compact Prediction Tree (CD-CPT). This modified algorithm is capable of exploring and outputting predictions based on the inputted sequence length for each new user's context events, generating multiple predictions for several new users in a single runtime, and exporting the results into a CSV file for further analysis. These enhancements make CD-CPT more suitable for predicting context data in real-world scenarios.

In this study, we leverage these existing algorithms, including TDAG (Transition Directed Acyclic Graph) and AKOM (All-k Order Markov) to generate short-term predictions based on current context events. We propose a modified method of CPT, called Context Data Compact Prediction Tree (CD-CPT), to predict real-world context data for multiple users. By comparing AKOM, TDAG, and other algorithms such as PPM, DG, CPT, and CPT+, we establish a majority baseline for context data prediction.

2.1 Sequence Pattern Mining Techniques

SPMF. To facilitate the implementation and comparison of CPT, CD-CPT, and other related algorithms, we employed the SPMF open-source data mining library. We opted for the Java version, as it provides access to additional algorithms and allows for customization of the code to obtain the desired output

AKOM and TDAG. All-k Order Markov and Transition Directed Acyclic Graph are sequence prediction models combining Markovian models of orders 1 to K . The techniques mentioned in this context find applications in diverse domains, such as natural language processing and image captioning. The K value is a user-adjustable parameter influencing the look-up window size and prediction accuracy. However, larger K values may consume more memory, making it less optimal than other algorithms with lower memory usage. In a separate study, AKOM, alongside the Long Short-Term Memory (LSTM) model(Hochreiter and Schmidhuber, 1997), Dependency Graph (DG), and Prediction by Pattern Mining (PPM), was found to be among the highest performance were utilized to forecast the upcoming three activities.(Tax, 2018).



II

Item	Seq_1	Seq_2	Seq_3	Seq_4	Seq_5
A	1	1	1	0	0
B	1	1	1	1	1
C	1	0	0	1	0
D	0	0	1	0	1
E	0	0	0	0	1

Figure 1: A prediction Tree (PT), Inverted Index (II) and a Lookup Table [2].

CPT. The Compact Prediction Tree (CPT) algorithm emphasizes lossless compression and low time complexity. Our modified version, CD-CPT, expands its capabilities to predict entire sequences of context data instead of single predictions. In Figure 1, the CPT algorithm's structure is illustrated, highlighting the use of a Prediction Tree (PT), an Inverted Index (II), and a Lookup Table (LT)(Gueniche et al., 2013). These components work together to enable efficient and accurate sequence predictions. In a different study, the CPT model was successfully applied to predict event sequences and enable quick learning in a robotic system, demonstrating the algorithm's adaptability and potential for various applications(Persia et al., 2020).

CPT+. The Enhanced Compact Prediction Tree (CPT+) algorithm is an upgraded version of the original CPT model that addresses limitations like time and space complexity. It stands out from other examined techniques due to its superior performance, efficiency, and its ability to be more compact and faster in predicting sequences.

CD-CPT: builds upon the original CPT model, enhancing its time and space complexity, which enables it to handle larger datasets and deliver more accurate predictions. This improvement distinguishes CD-CPT from other algorithms examined in this research, making it a powerful tool for predicting context data sequences across various scenarios.

DG. Dependency Graph (DG) is a sequence prediction model is used in this study due to its memory efficiency and ability to predict future symbols or events based on training sequences. It is compatible with the Java version of the SPMF library, allowing seamless integration with other SPMF algorithms used in this research. Originally designed for reducing user-perceived latency by predicting and prefetching files, DG offers good performance in analyzing context events.

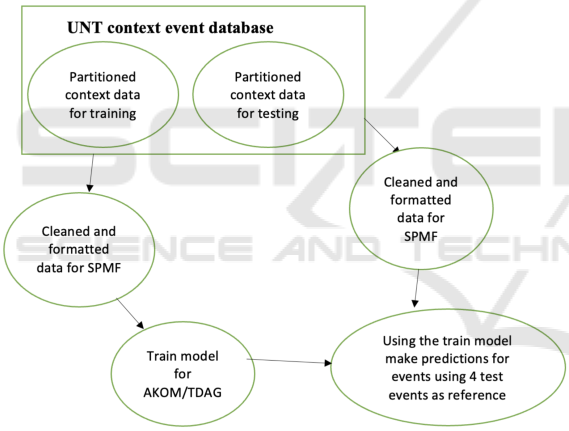


Figure 2: Research Methodology for AKOM and TDAG.

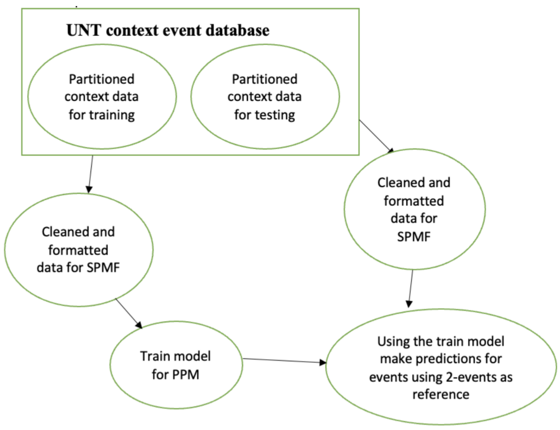


Figure 3: Research Methodology for PPM.

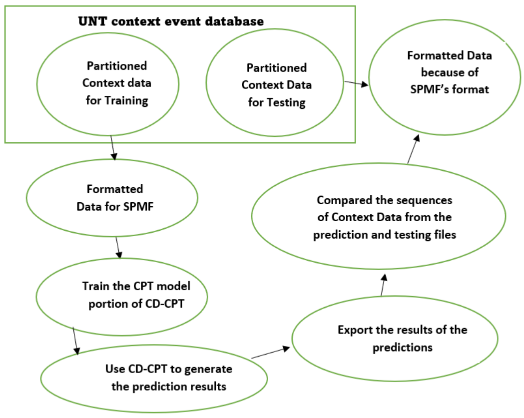


Figure 4: Research Methodology for CD-CPT.

PPM. The Prediction by Partial Matching (PPM) model, used in this study, is a fast and simple sequence prediction method, applicable to various fields. Although newer models like CPT+ may offer increased accuracy, PPM's versatility and adaptability make it a relevant choice for predicting context data. Despite the differences between context data and natural language, PPM delivered a fair score when processing and predicting the dataset in this research(Gellert et al., 2021).

Remote MySQL Database. A MySQL database combines context events from devices in the UNT context events database used in this study. It retrieves context events from a local SQLite database on a user's smartphone at 15-minute intervals and transmits the information to a remote server via HTTP and pushed it to MySQL database.

SQLite. SQLite is a built-in, serverless SQL database engine for the Android operating system . This software library allows the smartphone application to store context events locally on the device.

3 RESEARCH METHODOLOGY

Figures 2, 3, 4 to represent the research methodology for the AKOM, TDAG, PPM, and CD-CPT algorithms.

Figure 2 illustrates the research methodology for AKOM and TDAG, sequence prediction models known as context trees. These models combine Markovian models of different orders and adjust the input window size, represented by parameter K, to balance accuracy and memory consumption

Figure 3 illustrates the research methodology for PPM, which uses a sequence database for predictions. We experimented with parameters such as look-up

Table 1: PPM compared to relative algorithms' precision, recall, and F-1 score.

Algorithm	Support	Precision	Recall	F-1
PPM (order of 2)	**13939**	38.6	42.5	39.0
CPT (Original model) (window 5)	5575	8.6	19.7	10.6
CPT+(window 5)	5575	6.8	16.6	8.6
AKOM (order of 4)	6969	**55.4**	**55.2**	**54.4**
DG (order of 2)	**13939**	29.4	34.2	31.2
TDAG (order of 4)	6969	**55.4**	**55.2**	**54.4**
Scores that were produced by comparing the test files to AKOM and TDAG prediction's accuracy alongside the relative algorithms' prediction accuracy				

Table 2: Precision, Recall, and F-1 score from CD-CPT Model.

Precision	12.46
Recall	11.67
F-1	11.36
Total support size	27880

window, sequence window, and train file length to optimize performance. The look-up window, determining the number of previous events considered, is crucial in PPM.

Figure 4 depicts data extraction and preprocessing for the CD-CPT algorithm, which predicts context data sequences instead of single events. By analyzing key patterns and trends, we improved CD-CPT's ability to make accurate predictions.

4 EMPIRICAL STUDY

4.1 Evaluation Metrics

To evaluate the effectiveness of our approach, we developed two scoring systems for calculating precision, recall, and F-1 scores. These systems were created using Python with Google Colab for the CD-CPT algorithm and Java for the rest of the non-modified SPMF algorithms. Both methods provide accurate metrics based on our prediction models and test files.

4.2 Experimental Setup

The experiments compare the performance of CD-CPT, TDAG/AKOM,PPM, CPT+, CPT, and DG for prediction of real-world context events. The experiments in this study address the following research questions:

RQ1. What is the effectiveness of AKOM and TDAG compare to CD-CPT in predicting sequences of real-world context events, as measured by precision, recall, and F-1 score?

RQ2. What is the comparative effectiveness of AKOM and TDAG against other algorithms like CPT+, CPT, DG, and PPM in predicting real-world context events, as measured by recall, precision, and F-1 score?

RQ3. How does the accuracy of CD-CPT compare to that of AKOM and TDAG in analyzing the most frequently occurring context events in the test file?

Table 2 shows the performance of CD-CPT, model yields higher precision compared to recall, with scores calculated based on each context event's support size, leading to more accurate results. We further analyzed the most frequently occurring events in the 11 real-world context data sequences from users and CD-CPT's performance in predicting them as shown in Table 3. Conversely, we examined the least occurring context events in the 11 user sequences. Since these events did not appear in the test sequences, CD-CPT scores were set to 0. CD-CPT operates by predicting context events for a given number of users and their sequence lengths. After generating predictions, they are compared to the test file content to assess accuracy through precision, recall, and F-1 score. The performance of AKOM and TDAG, as shown in Table 4, demonstrates their ability to make good predictions compared to the majority baseline. Both models exhibit slightly stronger recall than precision. We further analyze the most and least frequently occurring events in the 11 real-world context data sequences from users and the performance of AKOM and TDAG in prediction. AKOM and TDAG significantly outperform PPM, CPT, CPT+, and DG as shown in table 1 Support size fluctuations between different algorithm models are due to specific look-up window sizes. PPM and DG have a small look-up window of 2, while AKOM and TDAG have a look-up window of 4, and CPT and CPT+ have the largest look-up window of 5. AKOM and TDAG use the first four context events from the test file to predict subsequent events and generate the support size.

Table 3: Most frequent Context Events for CD-CPT.

Event	Support	Precision	Recall	F-1
4 ('data_connection', 'lte_connected')	3232	12.77	16.18	14.27
22 ('audio', 'audio_effects_opened')	2880	28.97	24.72	26.68
21 ('audio', 'audio_effects_closed')	2738	32.79	26.66	29.41
6 ('data_connection', 'wifi_connected')	2557	10.25	14.12	11.88
2 ('configuration', 'changed')	2243	8.07	12.80	9.90
Scores produced by comparing CD-CPT prediction results csv file to the test csv file.				

Table 4: Most frequent Context Events for AKOM and TDAG using an order of 4.

Event	Support	Precision	Recall	F-1
21 ('audio', 'audio_effects_closed')	760	82.4	87.9	85.0
4 ('data_connection', 'lte_connected')	819	46.5	55.1	50.4
6 ('data_connection', 'wifi_connected')	653	51.4	64.3	57.1
22 ("'audio', 'audio_effects_opened')	633	87.1	89.4	88.2
2 ('configuration', 'changed')	586	48.1	46.6	47.4
Scores produced by comparing AKOM and TDAG prediction's alongside the test file.				

5 RESULTS AND DISCUSSION

RQ1. We compared AKOM and TDAG to CD-CPT using weighted averages for precision, recall, and F-1 score from 11 real-world context data sequences. CD-CPT's results as shown in Table 2 used 15 training sequences and achieved an F1-Score of 11.36% with a support of 27,880. AKOM and TDAG as shown in Table 1 used 47 training sequences and achieved a weighted F-1 score of 54.4% with a support of 6,969. The accuracy difference is significant; AKOM and TDAG outperformed CD-CPT but required test file reference. CD-CPT had lower scores but predicted entire sequences without referencing real-world context data. In summary, AKOM and TDAG excel in small event windows, while CD-CPT is better for predicting entire sequences without reference.

RQ2. Table 1 shows that AKOM and TDAG have similar performance and do better than the majority baseline when comparing their precision, recall, and F-1 scores. Their precision and recall stand at 55.4% and 55.2%, respectively, while the next best from the baseline (PPM) has 38.6% precision and 42.5% recall. The F-1 scores for AKOM and TDAG are 55.4%, with PPM having 39.0%. Different window sizes affect support: AKOM and TDAG use a window of 4 (support of 6,969), PPM and DG use a window of 2 (support of 13,939), and CPT and CPT+ use a window of 5 (support of 5,575). Despite support differences, the scores remained consistent when adjusting for normal count. In conclusion, AKOM and TDAG significantly outperform the majority baseline.

RQ3. To evaluate RQ3, we compared the highest supported context events for CD-CPT and AKOM/TDAG. CD-CPT's highest supported context event 'data_connection', 'lte_connected') had a support of 3,232 and a weighted F-1 score of 14.27%. In contrast, AKOM and TDAG had the same context event with a support of 819 and a weighted F-1 score of 50.4%. CD-CPT uses 15 sequences of user context events without referencing the test file, while AKOM and TDAG have constant access to the test file. The latter models have significantly lower support due to their window size of 4, which leads to discarded context events from the test file. This factor likely contributes to their higher F-1 scores. In conclusion, AKOM and TDAG achieve more accurate predictions with access to the test file, while CD-CPT generates decent predictions without referencing the test file.

In summary, RQ1, RQ2, and RQ3 offers guidance for researchers when considering the results of AKOM, TDAG, and CD-CPT. RQ1 highlights the suitability of each algorithm model depending on project requirements. RQ2 demonstrates that AKOM and TDAG outperform similar models, suggesting they are superior for short context event predictions. Lastly, RQ3 provides insight into the accuracy of AKOM, TDAG, and CD-CPT for specific context event predictions, which could be valuable for applications testing the likelihood of particular events occurring.

5.1 Threats to Validity

The users, user behaviors, and devices/apps that they used may not represent all users. We tried to minimize this threat by collecting data from 58 test subjects over a 30 day period. CD-CPT was tested on a limited set of 11 sequences of real-world context events from users and was optimized specifically for this dataset to improve its performance. These optimizations included adjusting the CPT-prediction scores for each predicted context event after it was predicted, encouraging the model to explore. Other optimizations involved adjusting the CPT prediction scores for other context events based on their probability of occurrence after specific context events.To mitigate these threats to validity, the data was cleaned for redundancy and transformed to ensure compatibility with the algorithm's input format. Additionally, future research may examine testing the model on larger and more diverse dataset to better assess generalization of this research.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we investigate various sequence prediction algorithms, such as AKOM, TDAG, PPM, DG, CPT, and CPT+, to predict real-world context data for smartphones, and propose a new method called CD-CPT (Context Data Compact Prediction Tree) for improved performance. The results show that AKOM and TDAG had the highest F-1 score of 54.4% with a look-up window of four, while PPM had an F-1 score of 39.0% and performed the best with a look-up window of two. CPT+ had a lower F-1 score of 8.6 compared to the other algorithms. CD-CPT, our proposed method, was able to predict sequences of real-world context data from users with an F-1 score of 11.36% using only the training model. Overall, the findings suggest that AKOM and TDAG are more accurate for single event predictions and CD-CPT was better at predicting full sequences of context data. Future work may use AKOM or TDAG during software testing to monitor different patterns of context events. Researchers may further investigate the use of CD-CPT for prediction and compare full sequences of context data from users. The study highlights the importance of choosing appropriate algorithms for predicting context data on smartphones, as this can significantly impact the performance and user experience of various applications

Future work may examine fault finding and effectiveness of integrating context event sequences into automated testing processes. Future work may also explore CD-CPT applied to domains such as smart watches, healthcare devices, various Internet of Things (IoT) devices and autonomous vehicles.

ACKNOWLEDGEMENTS

This work was supported by NSF grant #2149969.

REFERENCES

(2022). Android Releases — Android Developers.

Bosmans, S., Mercelis, S., Denil, J., and Hellinckx, P. (2019). Testing iot systems using a hybrid simulation based testing approach. *Computing*, 101:857–872.

Cleary, J. and Witten, I. (1984). Data compression using adaptive coding and partial string matching. *IEEE transactions on Communications*, 32(4):396–402.

Data.ai (12 Jan 2022). Number of Mobile App Downloads Worldwide from 2016 to 2021 (in Billions). *Statistica*.

Dey, A. K. (2001). Understanding and using context. *Personal and ubiquitous computing*, 5:4–7.

Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu, C.-W., Tseng, V. S., et al. (2014). Spmf: a java open-source pattern mining library. *J. Mach. Learn. Res.*, 15(1):3389–3393.

Fournier Viger, P. e. a. (2016). The SPMF Open-Source Data Mining Library Version 2. *Proc. 19th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2016) Part III*, pages 36–40.

Gellert, A., Precup, S.-A., Pirvu, B.-C., Fiore, U., Zamfirescu, C.-B., and Palmieri, F. (2021). An empirical evaluation of prediction by partial matching in assembly assistance systems. *Applied Sciences*, 11(7):3278.

Goyal, P., Khan, M. K., Steil, C., Martel, S. M., and Bryce, R. (2023). Smartphone context event sequence prediction with poermh and tke-rules algorithms. In *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0827–0834. IEEE.

Gueniche, T., Fournier-Viger, P., Raman, R., and Tseng, V. S. (2015). Cpt+: Decreasing the time/space complexity of the compact prediction tree. In *Advances in Knowledge Discovery and Data Mining: 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part II 19*, pages 625–636. Springer.

Gueniche, T., Fournier-Viger, P., and Tseng, V. S. (2013). Compact prediction tree: A lossless model for accurate sequence prediction. In *Advanced Data Mining and Applications: 9th International Conference, ADMA 2013, Hangzhou, China, December 14-16, 2013, Proceedings, Part II 9*, pages 177–188. Springer.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Laird, P. and Saul, R. (1994). Discrete sequence prediction and its applications. *Machine learning*, 15:43–68.

Mani, K. and Suneetha, K. (2020). Performance evaluation of compact prediction tree algorithm for web page prediction. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pages 1–7. IEEE.

Padmanabhan, V. N. and Mogul, J. C. (1996). Using predictive prefetching to improve world wide web latency. *ACM SIGCOMM Computer Communication Review*, 26(3):22–36.

Persia, F., D'Auria, D., and Pilato, G. (2020). Fast learning and prediction of event sequences in a robotic system. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 447–452. IEEE.

Piparia, S., Khan, M. K., and Bryce, R. (2021). Discovery of real world context event patterns for smartphone devices using conditional random fields. In *ITNG 2021 18th International Conference on Information Technology-New Generations*, pages 221–227. Springer.

Pitkow, J. and Pirolli, P. (1999). Mininglongestrepeatin g subsequencestopredict worldwidewebsurfing. In *Proc. UsENIX symp. on Internet Technologies and systems*, volume 1.

Rahmati, A. and Zhong, L. (2012). Studying smartphone usage: Lessons from a four-month field study. *IEEE Transactions on Mobile Computing*, 12(7):1417–1427.

Sarker, I. H., Colman, A., Han, J., Khan, A. I., Abushark, Y. B., and Salah, K. (2020). Behavdt: a behavioral decision tree learning to build user-centric context-aware predictive model. *Mobile Networks and Applications*, 25(3):1151–1161.

Sarker, I. H. and Salah, K. (2019). Appspred: predicting context-aware smartphone apps using random forest learning. *Internet of Things*, 8:100106.

Tax, N. (2018). Human activity prediction in smart home environments with lstm neural networks. In *2018 14th International Conference on Intelligent Environments (IE)*, pages 40–47. IEEE.