Discover User-App Interactions & Solutions to Reducing the Initial User-CPU Latency

Thy Nguyen Milon Chakkalakal Pranav Thaenraj

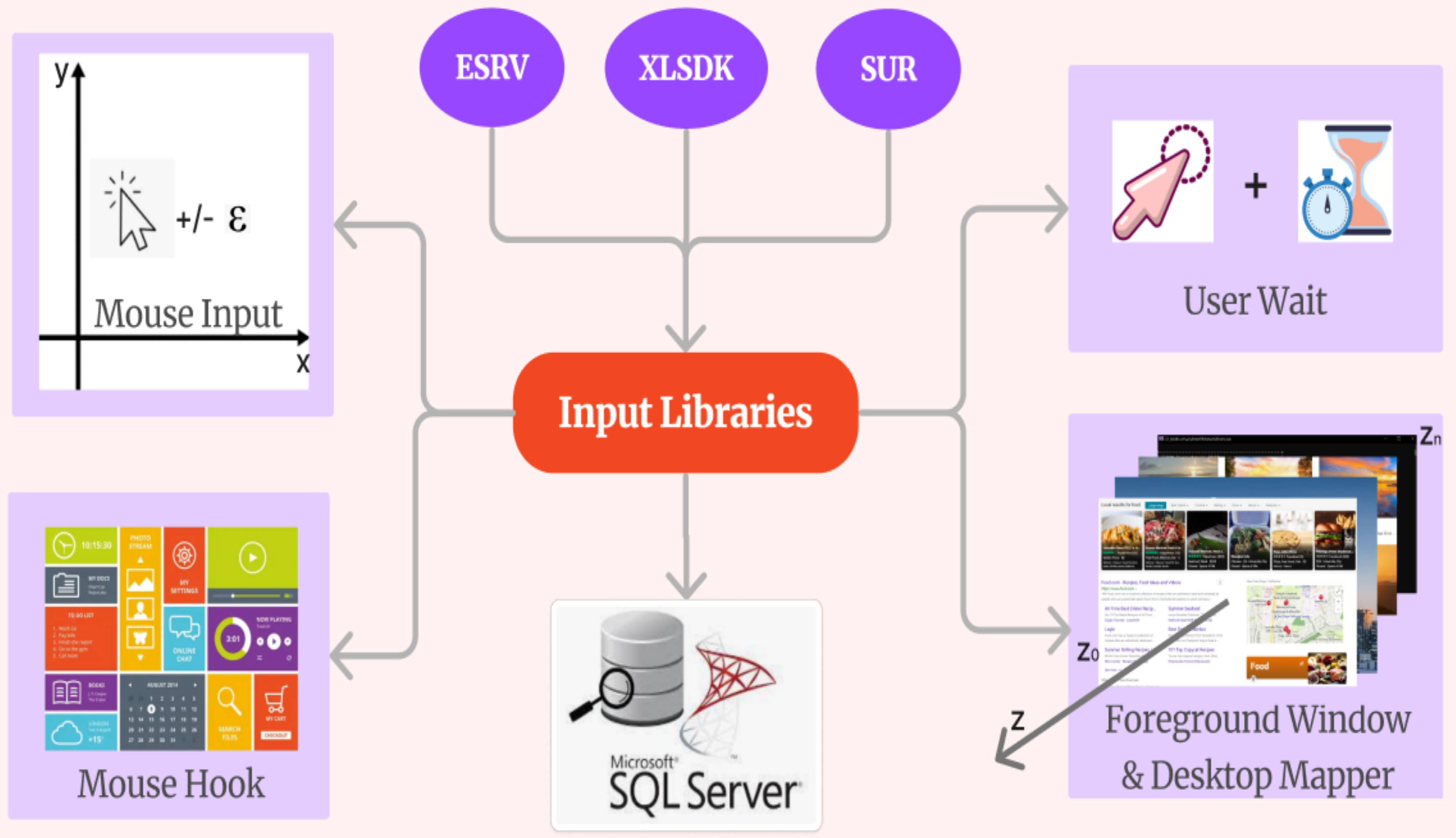
Advisors: Jamel Tayeb, Bijan Arbab, Sruti Sahani, Oumaima Makhlouk, Praveen Polasam, Chansik Im

Abstract

- “Data loading” icons signal unpleasant user-wait experiences and can tear people away from using an app.
- We mitigate the initial latency by collecting system usage data using Intel’s Telemetry and analyzing past behaviors by EDA, HMM, and LSTM/RNN.

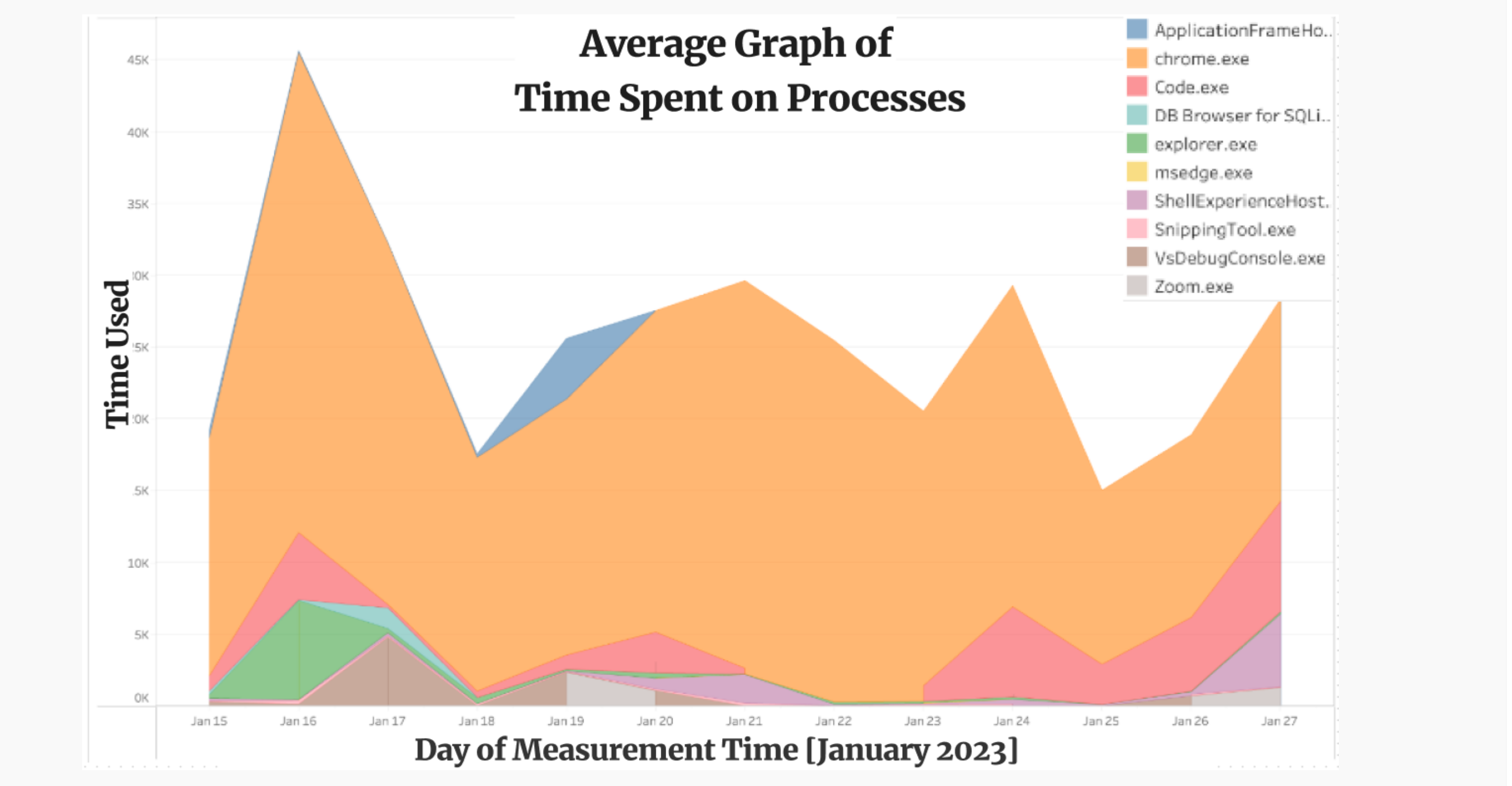
Methodology of Data Collection

- **Tools**: Software Development Kit, Environment Server, Intel® System Usage Report framework
- **Purposes** Anonymously gather and analyze data usage from multiple devices.



Exploratory Data Analysis

Chrome is the top frequently used app of this user according to the time measurement in 01/2023



Methodology of Predictive Tasks

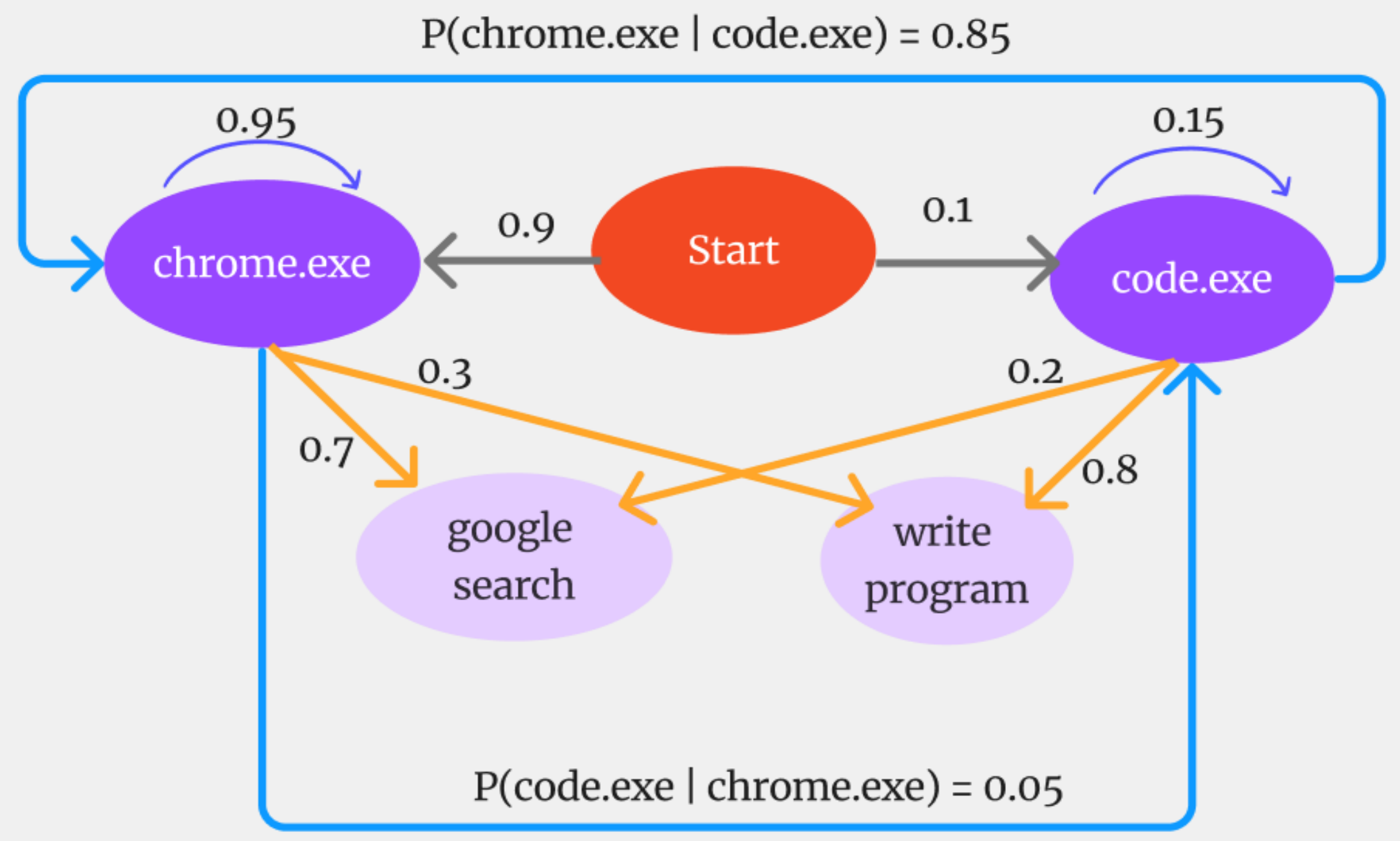
Hidden Markov Model (HMM)

- **Problem Statement**: Predict the likelihood of using an app *given* the former sequence of application usage
- **Idea**: Use conditional probability $P(A|B) = \frac{P(A \cap B)}{P(B)}$
- **A1. Markov Chain**: Only the current state q_{i-1} plays the most crucial role in predicting the future in the sequence

$$P(q_i = a | q_1 q_2 ... q_{i-1}) = P(q_i = a | q_{i-1})$$

- **A2. Output Independence**: The probability of observing an event o_i only relies on the state q_i that directly produced o_i

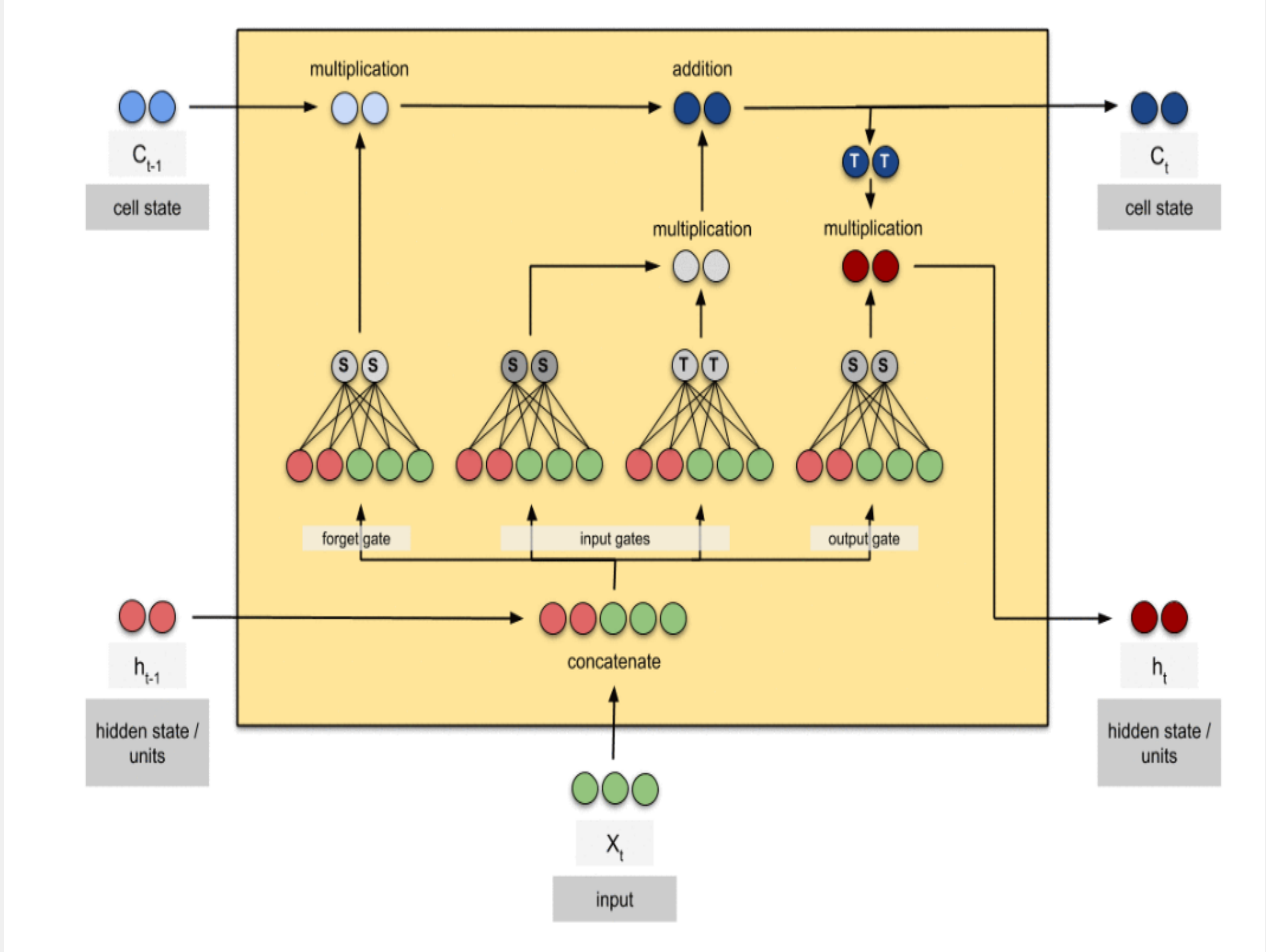
$$P(o_i | q_1, ... q_i, ..., q_T, o_1, ..., o_i, ..., o_T) = P(o_i | q_i)$$



- **Metrics**: Preds==True if within top n probabilities of the app

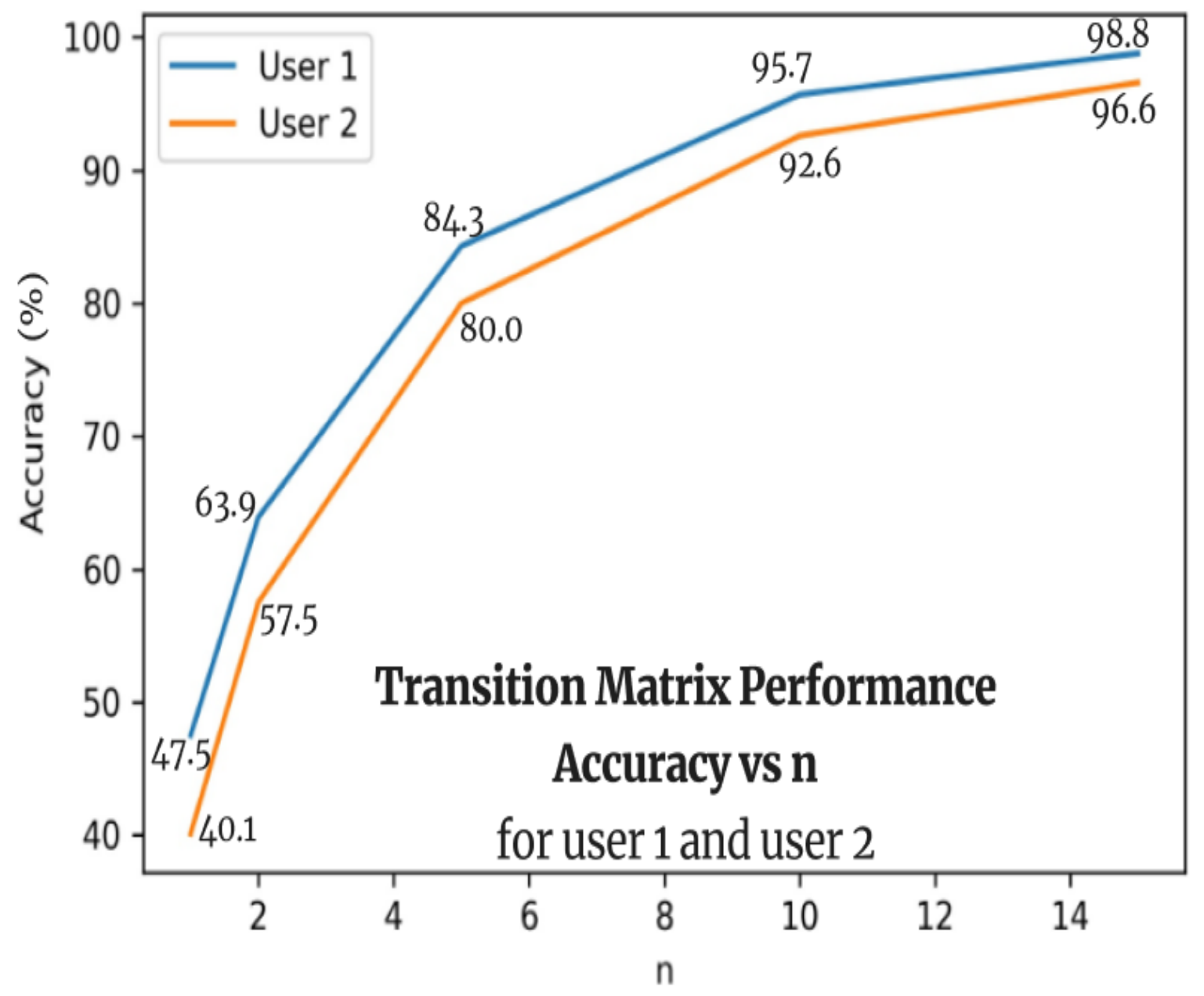
Recurrent Neural Network (LSTM/RNN)

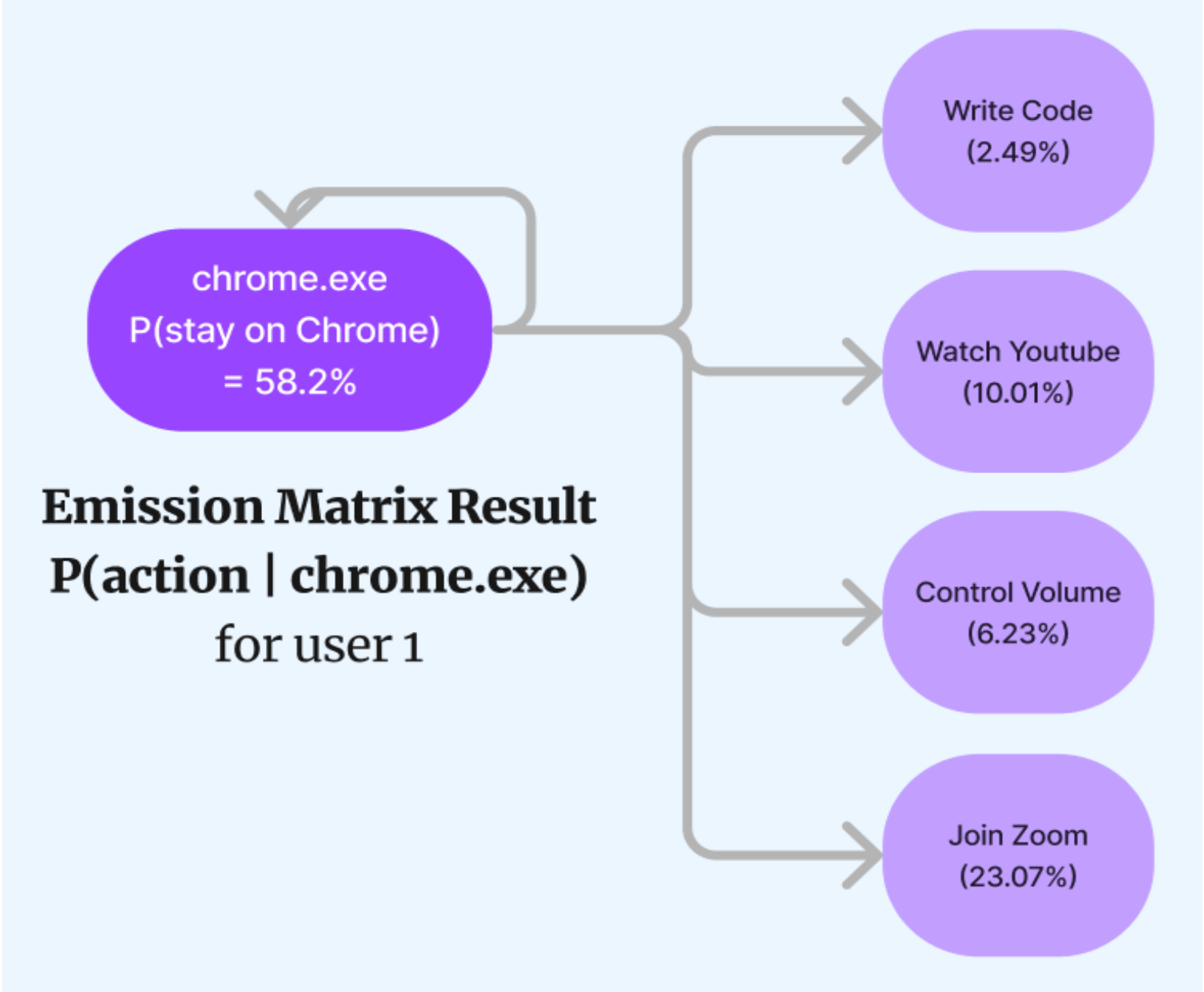
- **Problem Statement**: Predict the (total) time usage of an app/tab/recorded process using the past *time-series* data



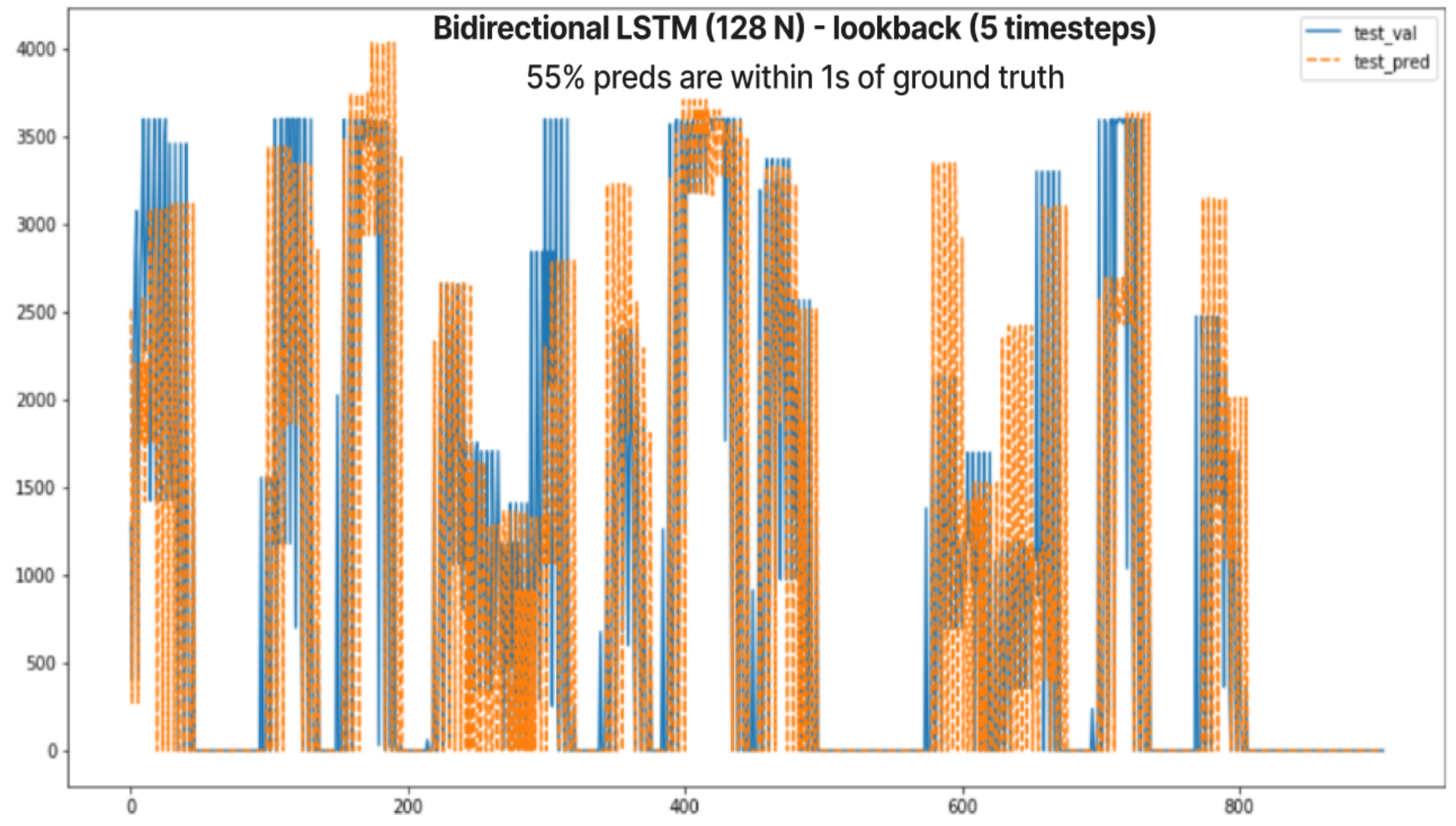
- **Feature Engineering**:
  1. Hourly split daily usage into 24 cols (labeled 0 - 23)
  2. Lookback 3-5 time steps from the current timestamp
  3. One-hot-encoding; Min-Max scaler
- **Experiments**: Train/Test: 80/20, no shuffle
- **Metrics**: RMSE, TP/TN/FP/FN, Preds==True if w/in 1 sec

Predictive Results





Emission Matrix Result
P(action | chrome.exe)
for user 1



Conclusion

- Data Collection
  1. *Foreground Window IL* provides the most applicable data for further analysis using EDA, HMM, RNNs
  2. Data should be collected *continuously* and *consistently* to allow detecting the patterns of user behaviors
- Predictive Tasks
  1. More data -> better predictive results
  2. HMM: The accuracy increases as n (the number of apps) increases
  3. RNNs: Feature engineering, input RNN libraries, and hyperparameter tuning greatly influence the model performance
  4. We can use these results to infer the daily sequence of app usage and their use times –> devise solutions to reduce the initial user-CPU latency
- Proposed Solutions

  Using our predictive results, we can further develop a script to process the