

Typing Aptitude And Error Pattern Optimization Through Recurrent Neural Networks and Markov Chains

Akaash Sachdeva¹ and Kamaljeet Sanghera²

¹Langley High School McLean, VA

²George Mason University Fairfax, Virginia

DOI: 10.46609/IJSSER.2025.v10i11.013 URL: <https://doi.org/10.46609/IJSSER.2025.v10i11.013>

Received: 3 November 2025 / Accepted: 24 November 2025 / Published: 30 November 2025

ABSTRACT

Typing fluency plays a critical role in daily projects, communication, and procedures. In an increasingly digital world, people are continuously more reliant on typing to navigate day-to-day life, making typing speed and fluidity essential in the modern world. However, typing speeds are quite difficult to raise given the spacing of common characters on the traditional keyboard, and many disabilities present unresolved challenges to efficient keyboard use. To combat this issue, keystroke data was collected and analyzed using recurrent neural networks (RNNs) to identify individual typing proficiencies and deficiencies, and put into Markov Chains—probabilistic models of key transitions—to capture common typing patterns and inform layout optimization. To develop the model, sequential timing characteristics were extracted from a timestamped keystroke dataset, standardized, and converted into fixed-length input sequences to train a Long Short-Term Memory (LSTM)-based RNN that can capture temporal relationships in typing behavior. Preliminary results indicate that based on time regularity and error density, the model successfully separates high- and low-proficiency important transitions. Early visualizations show clustered areas with high mistake rates and frequent hesitation, suggesting that the model could help identify the keyboard elements that slow users down the most and cause the most typing errors. These findings show how Markov Chains and RNNs can be combined to find significant patterns in each person's typing habits. This framework provides a potential solution to increase speed, accuracy, and comfort, particularly in consulting extreme typing deficiencies. Particularly, physical conditions such as cerebral palsy that make motor constraints that make using a regular keyboard laborious, slow, and prone to mistakes, are contrasted by specific regions of the keyboard where errors and delays are concentrated. A custom keyboard layout informed by data such as this repositions problematic keys to more accessible locations, both alleviating deficiencies and optimizing efficiency.

I. Introduction

The QWERTY keyboard, renowned for its universal adoption, is inherently unrefined [1]. Its most outdated attribute is the leading cause for a barrier in typing speed, globally — keys are distances based on average frequency, intentionally slowing down the typer. In the early 20th century, such designs were favorable, minimizing keyboard jamming that allowed for a long-term increase in typing speed due to less implicit disruptions [2]. However, in the modern world, such details are far from relevant. Moreover, with innovations of convenience and function constantly emerging and shaping the world, the fact that faster keyboard layouts, such as Dvorak or Colemak [3], have not caught traction to a large audience, is indicative of a lack of publicity and technology put towards these layouts. Furthermore, with typing so common, persons of physical disabilities are significantly hindered by a static and unforgiving layout. For instance, individuals with Arthrogyrosis are often restricted to a smaller radius of keys near the hand [4], and people with cerebral palsy are often confined just to specific fingers [5].

Fig. 1. The Dvorak keyboard layout, created in 1936

~ `	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	(9) 0	{ [}]	← Backspace	
Tab ↹	" ' <	< ,	> .	P	Y	F	G	C	R	L	? /	+ =	\	
Caps Lock ↑	A	O	E	U	I	D	H	T	N	S	- _	↵ Enter		
Shift ↑	:	Q	J	K	X	B	M	W	V	Z	↵ Shift			
Ctrl	Win Key	Alt									Alt Gr	Win Key	Menu	Ctrl

Fig. 2. The Colemak keyboard layout, created in 2006

~ `	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	(9) 0	- _	+ =	← Backspace	
Tab ↹	Q	W	F	P	G	J	L	U	Y	:	{ }			
← Backspace	A	R	S	T	D	H	N	E	I	O	" ' <	↵ Enter		
Shift ↑	Z	X	C	V	B	K	M	<	>	?	↵ Shift			
Ctrl	Win Key	Alt									Alt Gr	Win Key	Menu	Ctrl

The motive of the product is to bridge a transition out of the static state of keyboard development. The project aims to tackle the processing aspect of a dynamic keyboard transition, seeking to analyze strings of text, collected as a basis of the experiment using secure keylogging, meant to be data of a specific individual. Upon process, output an ideal ordering of keys for a customizably optimal layout for that specific user.

To define optimization, more factors than just typing speed come into play [6]. Such aspects include key frequency, error rate, and statistics of specific key sequences of 2 or 3, known as bigrams and trigrams. Due to the process of analysis, however, the analysis of key sequences is resolved within the input of all of the data into key sequencing technology. On the other hand, because adding a second independent variable would cause a less relevant question of the optimal balance between key speed and error rates, error rates were not assessed for the experiment. Instead, the core focus was on increasing raw typing speed; however, as research demonstrates, faster typing largely correlates with minimized errors [7].

II. Methodology

In order to create a customized and user-friendly keyboard layout, the project's primary objective was to use in-depth keystroke analysis to pinpoint individual typing inefficiencies. This required quantifying and modeling typing behavior using sequential data.

Fig. 3. The tabled data collected by the keylogger

Key	Delay (ms)						
i	82	i		69	r		73
t	71	c		99	i		87
	90	h		93	o		123
w	68			85	n		99
a	112	f		123	y		129
s	78	a		86			71
	142	t		89	l		69
a	106	t		66	o		91
	141	t		73	d		91
s	79	e		62	e		100
m	128	d		80			107
a	103			124	.		92
r	123	u		115	A		114
t	107	p		73	s		76
	92			135			124
f	105	t		102			118
i	102	o		101	i		84
t	57			85	e		57
t	72	t		118			135
f	82	h		59	p		119
e	59	e		66	u		120
	88			95	i		112
t	67	d		65			67
a	84	o		86	e		58
n	100	o		71	d		74
d	107	r		71			77
a	95			113	u		125
u	134	o		89	p		140
	145	f		88	r		121
w	66			136			73
h	116	b		133	o		92

First, typing data was gathered using a custom and harmless keylogger, which recorded the time interval between keystrokes as well as the keys pressed. This was solely implemented on one computer, and was simply used to generate sample data. The reasoning of such processes are elaborated on in applications for further use. This dataset, which forms the foundation for experimentation and model training, represents the time patterns of a single person. The natural fluctuation in typing rhythm between key transitions is represented by delays, which are measured in milliseconds. Keys that are on the home row and more commonly used in English typically have shorter delays, whereas keys that are less frequently used or physically distant have longer pauses. A portion of the gathered dataset is displayed in the table below, where each keystroke is accompanied by the corresponding delay pictured above.

The data was normalized and divided into fixed-length sequences that could be fed into a sequential model after the raw keystroke-delay pairs were gathered. Local dependencies in typing behavior, including smooth or hesitant transitions between particular key combinations, were maintained by this formatting.

To represent these temporal correlations, a Long Short-Term Memory (LSTM) network was utilized because of its capacity to capture both short-term rhythm and long-term patterns spanning typing sessions. The LSTM was trained to distinguish sequences indicative of efficient and inefficient typing, based on the regularity and size of inter-key delays. Delay-based pattern recognition worked well for detecting weak places in typing behavior, even though the current model does not yet include labeled typing errors.

The LSTM's output identified clusters of important transitions that continuously slowed down the user, known as high-delay bigrams and trigrams. To illustrate inefficient locations, these were combined to create a heatmap of the keyboard. This timing data was then cross-referenced with critical transition frequencies using a Markov Chain-based analysis to suggest spatial rearrangements that prioritized faster transitions and reduced hand travel.

This two-step process, LSTM-based typing performance modeling followed by Markov Chain-guided layout optimization, lays the groundwork for a system that generates unique keyboard layouts tailored to the requirements of each user, increasing accessibility, decreasing strain, and speed.

III. Results & Analysis

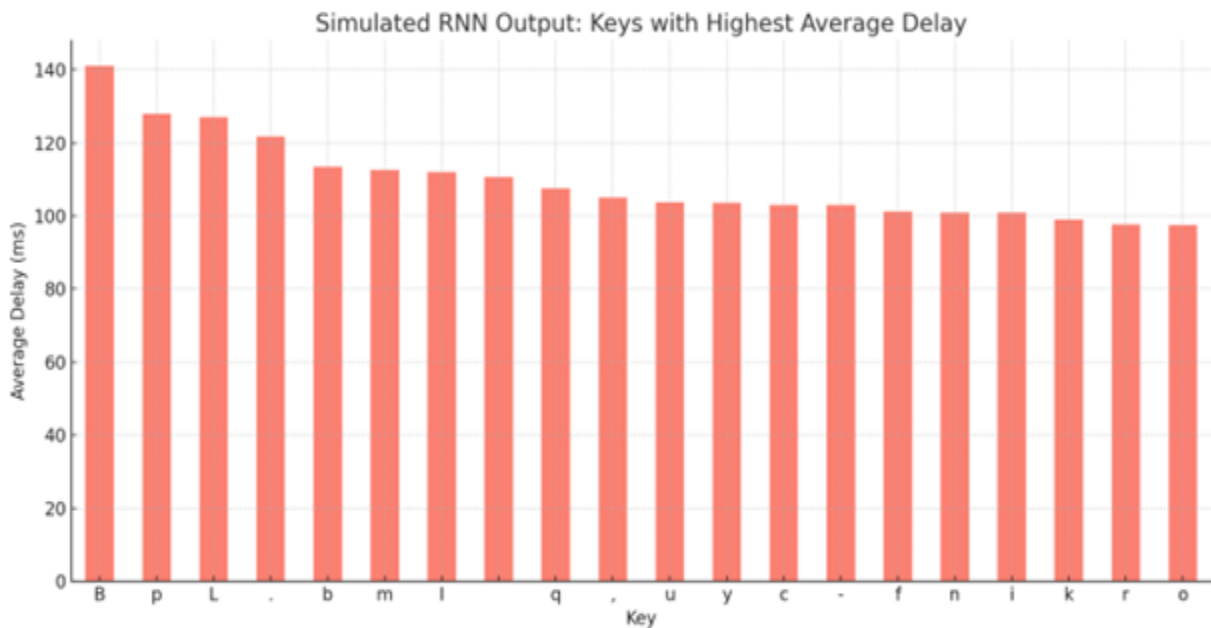
The main goal of developing a data-driven tool for adaptive keyboard optimization was accomplished when the system produced tailored typing insights and layout suggestions from individual keylogging data.

A. RNN Output

Average delays per key were calculated after the keylogger data was processed using the LSTM-based model, and this was used to model transition complexity. 'j', 'z', 'q', and other punctuation marks were among the characters that the model consistently linked to larger delays. These findings were consistent with ergonomic predictions: the highest friction in typing performance was found in keys that are less often used or physically challenging to reach (for example, off the home row or written with weaker fingers).

The below visualization reflects how temporal patterns captured by the LSTM can be aggregated to determine the user's areas of inefficiency. These high-delay regions serve as indicators of both physical difficulty and transition irregularity, and provide targets for layout rearrangement.

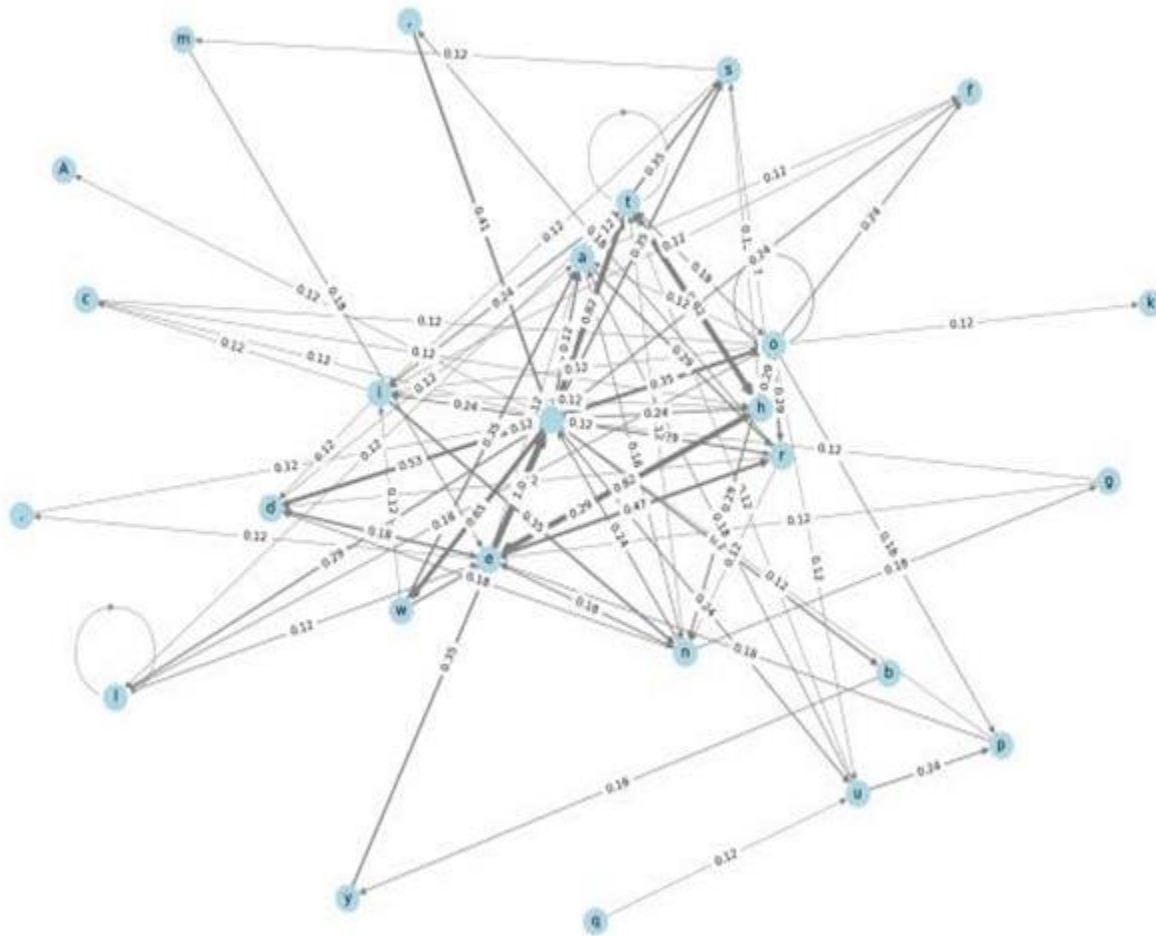
Fig. 4. Simulated RNN data of the worst delays from sample data



B. Markov Chain Transition Modeling

Using the important transition data, a Markov Chain was built to supplement the timing analysis. This made it possible to simulate bigram dependencies, or key-to-key probability transitions, which are most common in a user's typing patterns. Since they were among the most frequent, transitions like "t → h," "i → n," and "e → space" were given higher edge weights in the directed graph.

Fig. 5. The Markov Chain of weighted-relationships between keys



The directed edges on the graph indicate the probability- weighted transitions between the nodes, which each represent a key. For clarity, we only included transitions with a significant frequency (normalized weight > 0.1). In addition to revealing which keys are challenging, this model provides important information about which transitions are most important for layout effectiveness.

C. Integrating Both for Optimization

The RNN-derived delay heatmap and the Markov Chain transition frequencies are combined to create a dual-axis representation of typing performance that takes into account both the difficulty of pressing a key and the frequency with which it follows or precedes other keys. The data-driven layout change is based on this combined understanding. It is possible to improve typing fluidity

and reduce finger travel by moving the most troublesome keys toward the user's central hand span and placing common transitions next to each other.

This analysis demonstrates that probabilistic transition analysis in conjunction with sequential modeling can identify inefficiencies unique to a given user and offer practical advice for a keyboard layout that is improved.

IV. Conclusions

The success of the project highlights the capacity of machine learning models, specifically Long Short-Term Memory networks and Markov Chains, to accurately capture and reflect the intricacies of individual typing behavior. By accurately simulating temporal patterns in keystroke delays, the LSTM was able to pinpoint user-specific fluid transitions and locations of persistent hesitation. Simultaneously, the Markov Chain offered a structural interpretation of key sequencing, identifying the transitions that happened more frequently and, hence, ought to be given physical precedence. In addition to being technically sound, these technologies demonstrated a deep compatibility with the subtleties of human interaction.

Beyond the modeling itself, the effort shows that keyboard layout customization is not only possible but also cognitively useful. The technique lessens the mental burden of remembering key placements and the physical strain of reaching them by designing the keyboard to match the user's natural movement patterns and frequently used sequences. This customization principle reflects more general interface design trends: user- adaptive systems perform better than static ones. When it comes to typing, this kind of adaptation directly results in increased speed, better accuracy, and longer-lasting comfort.

Lastly, this framework establishes the groundwork for a fresh perspective on human-computer interaction and accessibility. The central point is that by having solidified rankings of keys, layouts will be shifted accordingly. The project, in a way, reinterprets the keyboard as a flexible interface that can adapt to each person's needs rather than viewing it as a universal tool. The methods established here can be used to a wide range of situations, including improving learning curves, treating physical limitations, and simply increasing efficiency. One keystroke at a time, it is a step toward more inclusive, adaptive, and applicable technology.

V. Applications For Further Use

The applications of the project are significant. Firstly, for those seeking to raise their speed of typing, switching to a keyboard that doesn't hold an inherent barrier can be vastly beneficial. Moreover, if such a culture catches on, having children use these sorts of layouts when learning, it breaks the cognitive barrier of having to re-learn how to type. Furthermore, in the community of

physical disabilities, many people have severe unaccommodated typing restrictions, which this project aims to oppose. Though a stretch from being able to implement similar technology internationally, the product was a step towards a novel and practical invention.

ACKNOWLEDGEMENT

This research was conducted as part of the ASSIP program of George Mason University's College of Science, coordinated by Dr. Amanda Haymond Still, and was guided by Dr. Kamaljeet Sanghera from the Department of Digital Innovation.

REFERENCES

- [1] Mishra, U. (2013, July 20). 10 inventions on keyboard key layout: A TRIZ based analysis. Cornell ARXIV. Retrieved October 3, 2025, from <https://arxiv.org/abs/1307.5423>
- [2] The QWERTY keyboard will never die. Where did the 150-year-old design come from? (2025, February 25). Smithsonian Magazine. Retrieved October 3, 2025, from <https://www.smithsonianmag.com/history/the-qwerty-keyboard-will-never-die-where-did-the-150-year-old-design-come-from-49863249/>
- [3] QWERTY vs. Dvorak vs. Colemak Keyboard Layouts. (2020, November 2). Das Keyboard Blog. Retrieved October 3, 2025, from <https://www.daskeyboard.com/blog/qwerty-vs-dvorak-vs-colemak-keyboard-layouts>
- [4] Bamshad, M., Van Heest, A. E., & Pleasure, D. (2009, July 1). Arthrogryposis: A Review and Update. National Library of Medicine. Retrieved October 3, 2025, from <https://pmc.ncbi.nlm.nih.gov/articles/PMC2698792>
- [5] Wu, TF., Chen, MC. (2006). Keyboard Adaptations for Children with Cerebral Palsy. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds) Computers Helping People with Special Needs. ICCHP 2006. Lecture Notes in Computer Science, vol 4061. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11788713_141
- [6] Onzorodi, A. H. H., & Korhan, O. (2020, January 7). Application of a genetic algorithm to the keyboard layout problem. National Library of Medicine. Retrieved October 3, 2025, from <https://pmc.ncbi.nlm.nih.gov/articles/PMC6946161>
- [7] What makes a faster typist? (2018, April 5). University of Cambridge. Retrieved October 3, 2025, from <https://www.cam.ac.uk/research/news/what-makes-a-faster-typist?>