Personalized Font Generation using Keystroke Dynamics

by

Narges Sayah Dehkordi
B.Sc., Amirkabir University of Technology, 2019

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

University of Victoria

We acknowledge and respect the Lək$^{\text{w}}$əŋən (Songhees and X sepsəm/Esquimalt) Peoples on whose territory the university stands, and the Lək$^{\text{w}}$əŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Personalized Font Generation using Keystroke Dynamics

by

Narges Sayah Dehkordi
B.Sc., Amirkabir University of Technology, 2019

Supervisory Committee

Dr. Miguel Nacenta, Supervisor
(Department of Computer Science)

Dr. Regan Mandryk, Committee Member
(Department of Computer Science)

ABSTRACT

Before the advent of digital communication, personal correspondence was often handwritten, allowing people the opportunity to express themselves in their own unique style. As digital communication has become more common, typed text often lacks the personal touch that handwriting conveys. Despite the wide variety of styles in modern font design, digital font uniformity limits individual identity in typed communication. The act of typing itself, however, is a nuanced activity with distinct patterns unique to each individual. This study explores how these unique typing patterns can be leveraged to generate personalized fonts, offering a form of digital self-expression similar to handwriting.

We present a system that analyzes keystroke dynamics, such as Keydown-Keydown time, Flight Time, and the spatial distribution of keys, to create customized fonts that are stable for individual participants yet unique across different participants. Using datasets from multiple universities, we preprocess and analyze typing behaviours, extracting features that are both highly discriminative and consistent. These features are then used to generate personalized fonts that visually reflect each participant's distinct typing style. Our system demonstrates the feasibility of personalized digital communication through typing behaviour-driven font generation, offering an innovative way to enhance individuality in electronic communications.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My mother, Roya, my father, Abas, and my sister, Niloufar, for being the constant thread of strength and grounding throughout every phase of this journey.

My aunt, Anita, whose support and presence during the final stretch of this thesis gave me the focus and comfort I needed.

My supervisor, Miguel Nacenta, for his thoughtful guidance, trust, and honest feedback.

The HCI research lab at UVic for the inspiring conversations and collaborations.

My friends, both near and across time zones, who reminded me to pause and keep perspective.

And finally, the beautiful city of Victoria, which gave me the space and clarity to do this work.

This thesis would not have been possible without them.

Chapter 1

Introduction

Over the last century, the widespread digitalization of communication has significantly shifted our reliance from handwritten correspondence to typed text. Despite the abundance of fonts available, many individuals feel that typed text often lacks the personal touch and expression naturally found in handwriting [35]. The subtleties of stroke thickness, letter spacing, and slant in handwriting contribute to its distinctiveness, offering a visual signature that is often absent in standard digital fonts. This loss of individuality can be particularly noticeable in personal communications, such as emails or digital journals, where the uniformity of typed text fails to convey the nuances of the writer's style.

While handwriting not only conveys a message but also reflects the writer's personality, mood, and even speed [36], typed text has remained disconnected from these personal characteristics. However, just as handwriting is unique to each individual, so too are typing dynamics. The way a person types, whether in terms of speed, key press duration, or rhythm, carries distinct patterns that, much like handwriting, have the potential to reflect personal traits, suggesting that keystroke dynamics could be harnessed to personalize the appearance of the content, and therefore offer an opportunity to bridge the gap between impersonal electronic text and the expressive qualities of handwriting.

This research explores how keystroke dynamics can be leveraged to create personalized fonts, thereby reintroducing individuality to digital communication. By examining patterns in typing behaviour, such as the timing between key presses, typing consistency, and other temporal aspects, we propose a system that transforms these behavioural signatures into unique font variations. This system aims to mimic the expressiveness of handwriting while preserving the practicality and efficiency of typed text.

To achieve this, we utilize pre-gathered datasets of keystroke dynamics from 228 participants to identify features that are the most distinguishing between individuals. By applying the Sum of Standardized Mean Differences (SSMD), we can pinpoint the most discriminatory features across participants. Once these distinguishing features are identified, we then analyze the typing data of the current participant to find the most stable features for that individual, which act as their behavioural signature. This step is

performed using the Coefficient of Variance (CV) to ensure the consistency and reliability of the features over time.

By combining two scoring mechanisms, one to identify inter-participant distinguishing features and another to select the most stable intra-participant features, we derive the values for specific font parameters that are most perceivable, such as stroke width, slant, and letter spacing. These values are then used to generate and render a fully custom font that reflects the unique typing behaviour of the participant. This ensures that the system is adaptable to a wide range of typing styles and stable enough to maintain consistent personalization over time. Our system does not merely generate a one-time custom font; instead, it adapts as a user's typing habits evolve, ensuring a dynamic and ongoing relationship between their typing and the visual output of their text. The resulting font can be downloaded through the system's user interface, allowing users to apply their personalized font to their digital communications. Throughout this thesis, we distinguish between the terms user and participant to define these roles within the context of our research. A participant refers to any individual whose keystroke data contributes to the evaluation of our font-generation pipeline, including individuals whose data were gathered externally or entered directly into the system interface for evaluation. In contrast, a user is an individual who actively and intentionally engages with the system to generate personalized fonts, knowingly and deliberately using it to create personalized fonts. This distinction highlights that participants may not necessarily have had intentionality influence their typing behaviour, unlike users who consciously adapt their behaviour knowing it affects visual font output.

1.1 Thesis Scope

This research aims to develop a prototype for personalized fonts generated through keystroke dynamics, designed to reintroduce individuality and expressiveness to digital communication. Our system examines how typing behaviour can be translated into visual characteristics, providing users with a unique way to represent themselves in typed form. This concept demonstrates the feasibility of using such a system in personal journals, casual communication, and any other informal digital context where a more humanized form of electronic communication, achieved by adding a personal touch to the text, can enhance the user experience.

Although the research focuses on generating fonts that reflect individual typing patterns, the resulting outputs may find natural use in personal digital contexts. For example, by applying personalized fonts to digital journals, users can create a unique and intimate feel, adding a reflective quality to their entries. Similarly, in casual text-based communication with friends, personalized fonts could offer an additional layer of expression, helping users convey aspects of their personality that might otherwise be lost in

uniform typed text.

However, the scope of this research is limited to generating personalized fonts that are distinct across users and remain consistent for the same user, provided their typing behaviour does not change significantly. We do not prioritize the readability of the generated fonts, as the primary focus is on capturing individual typing dynamics rather than ensuring that the output font is suitable for formal or professional contexts. Readability may vary between generated fonts, and some variations could potentially compromise text clarity, depending on the individual's typing characteristics. Thus, while the system provides unique and personalized output, it may not meet the stringent readability standards required for professional documentation or widespread publication.

It is essential to clarify that this system does not provide user identification capabilities. Although keystroke dynamics has been extensively used in research on authentication and user identification (often serving as an added layer of security), our system is not designed for these purposes. Instead, it is oriented toward personalization and self-expression. The generated fonts do not carry the capability to verify or authenticate user identity; they are merely reflective of typing patterns in an artistic, non-secure context. The emphasis is on visual uniqueness rather than on using keystroke dynamics as a biometric marker for security.

In summary, the scope of this research is to explore and validate the concept of individualized fonts based on typing behaviour, specifically in informal digital communications. While we aim to assess the value and feasibility of personalized digital text, considerations such as font readability and user authentication fall outside the scope of our intended focus. Through this exploration, our goal is to bring a new level of personalization to digital communication, highlighting the potential of keystroke dynamics to enrich digital self-expression.

1.2 Contributions of this Thesis

This thesis makes the following three contributions:

First, it proposes a new type of relationship between how text looks and how it is generated by a person. By linking keystroke dynamics to visual typographic output, the work reconnects the action of writing text with the variety of how it appears visually. This framing offers a new perspective on how behavioural input can influence text appearance, introducing a personal and continuous connection between writing and form.

Second, it demonstrates how an analysis of existing typing data can produce suitable parameters for typographic variation. Using pre-collected datasets, the thesis identifies features that are both stable within participants and distinctive across participants. These features are then used to drive specific font parameter values, showing that reliable and repeatable visual outcomes can be obtained from behavioural data alone.

Third, it contributes a prototype system that allows people to experience this mapping interactively. The system processes live typing input, computes the necessary feature values, and applies them to font parameters that can be seen in a few seconds. Users receive an on-screen preview of the text rendered in their personalized font. They can download the results, making the connection between typing behaviour and visual outcomes directly observable and testable.

1.3 Overview of the Following Chapters

The remainder of this thesis is structured as follows: Chapter 2 provides the theoretical background and related work, introducing key ideas such as InfoTypography, biometric typing patterns, digital font technologies, and statistical measures used in our analysis. Chapter 3 outlines the methodological framework and design principles that shape the system, establishing the rationale for each design decision. Chapter 4 introduces the keystroke datasets that underpin our work and explains how they are prepared for analysis. These datasets are then fed into the full system pipeline, which is described in Chapter 5, encompassing feature extraction, parameter mapping, and font rendering. Chapter 6 presents an evaluation of this implementation, considering how well the system captures individual traits and produces visually distinct fonts for users. Finally, Chapter 7 discusses the broader implications of the results, reflects on its limitations, and identifies directions for future research. Together, these chapters form a cohesive path from concept to prototype to evaluation, demonstrating the feasibility and expressive potential of behaviour-driven typography.

1.4 Use of Generative AI in this Thesis

Generative AI was utilized during the development of this thesis, specifically for proofreading assistance, grammatical accuracy, and enhancing the overall quality of the written text. Additionally, generative AI was occasionally employed for debugging purposes during the implementation phase of the font-generation system. However, it was not used to directly produce code, conduct analyses, or generate original research content.

Chapter 2

Background & Related Work

This chapter introduces the key areas of background and related work relevant to this thesis. There are three fundamental pieces of prior work that will provide the conceptual and technical foundation for the system proposed here: (1) research on typing patterns and keystroke dynamics as a form of behavioural signature; (2) developments in typography and font generation, especially methods for dynamic and personalized type design; and (3) statistical measures used to quantify the reliability and distinctiveness of behavioural data. Each of these areas is presented in the following sections.

2.1 Keystroke Dynamics

A prominent use of keystroke dynamics is to analyze typing patterns as a form of behavioural biometrics to identify or authenticate users based on timing features extracted from key events [39]. Unlike physical biometrics (e.g., fingerprints), keystroke patterns offer unobtrusive software-based identification using standard keyboards. Two primary modes exist: fixed-text dynamics analyze repeated inputs like passwords, while free-text dynamics assess arbitrary typing during ordinary use [5]. Temporal features commonly used include Down-Down (DD) time between key presses, Down-Up (DU) key hold time, and Up-Down (UD) latency between key releases and subsequent presses. Fixed-text approaches typically achieve higher accuracy due to consistent input, whereas free-text methods enable continuous monitoring throughout a session. A wide range of statistical and machine learning techniques has been applied to build participant profiles from these features, yielding promising authentication results in both modes. Comprehensive surveys of the field detail how these techniques and performance trends have evolved over several decades [55, 49], classifying research into static vs. dynamic input scenarios, feature representations, and classification algorithms.

Beyond authentication, keystroke dynamics have also been studied for other purposes. Researchers have leveraged typing patterns to infer participant characteristics or demo-

graphics. For example, distinguishing male vs. female participants based on typing and mouse behavior [32], as well as providing context-sensitive solutions. In affective computing, typing rhythm has been used to recognize emotional states: for instance, Epp et al. demonstrated that classifiers could identify several emotional states (such as confidence, nervousness, sadness, etc.) from keystroke timing with around 80% accuracy [17]. Keystroke dynamics have even been explored as indicators of cognitive load and mental state; recent work suggests that specific typing patterns correlate with stress, attention, or cognitive decline [43]. (These approaches treat keystrokes as a rich sensor of human state rather than as a security credential.) While there is not enough space here to detail all these directions, comprehensive surveys such as those by Teh et al. [55] and Shadman et al. [49] provide a deeper exploration, classifying existing research into several main categories. These existing studies are informative for us; they are interesting and valuable. However, the purpose of our research is fundamentally different from most previous studies in the field of keystroke dynamics. In particular, our work investigates keystroke dynamics as a drive for real-time visual personalization (font generation) rather than identity verification or user state detection per se.

2.2 InfoTypography

InfoTypography refers to the use of typographic variations, such as letter weight, size, slant, or spacing, to communicate more information than literal content. Historical examples include the use of bold or italic styles to emphasize or structure text. More recently, InfoTypography has been explored within visualization and HCI as a medium for data encoding, where letterforms themselves visually reflect data attributes [34, 1].

For example, FatFonts embed numerical values into digit shapes by modulating their fill weight: larger quantities appear as heavier, more filled-in numerals [41]. This technique bridges numerical and visual representations, allowing the text itself to function as a chart without losing legibility. Other researchers have proposed mapping secondary data variables onto the typographic parameters of text used in visualizations or user interfaces. Brath and Banissi, for instance, have systematically expanded the design space of text in visualizations, using font attributes like height, weight, or serif presence to encode data attributes in labels and passages [8, 9]. Their work and related efforts demonstrated that specific font parameters (e.g., stroke thickness, slant angle) are highly perceivable channels [2], making them suitable for conveying additional information such as statistical values or categorical distinctions.

These examples do more than highlight or structure the text; they use typography to encode extra information. As Nacenta describes, “the idea is that there’s some information that’s not there, something that adds a different dimension” [41], building on Bertin’s concept of visual variables, where font features like size, weight, and orientation

are treated as meaningful visual encodings.

Within HCI, InfoTypography opens avenues for personalization and self-expression. Prior work in this area focused mainly on data-driven visualization and perceptual optimization (e.g., ensuring infotypographic encodings are distinguishable [2], or exploring multivariate typographic theming on maps [9]). Brath's recent book Visualizing with Text surveys a wide range of techniques and applications where textual elements carry data or context beyond their literal meaning [7], underlining the rich potential of this approach.

In summary, InfoTypography treats the appearance of text itself as a medium for displaying information. Several compelling examples from the literature have informed our thinking. However, our use of InfoTypography is novel in that it leverages keystroke dynamics to customize typographic parameters for each user.

2.3 Font Generation

Digital font generation encompasses the creation of typefaces through algorithmic or parametric methods, allowing designers to produce variations of a font beyond the static styles of traditional outline fonts. Outline font technologies (e.g., TrueType, PostScript) are well-established but inflexible, since each style (bold, italic, condensed, etc.) must be a separate font file and continuous variation is not supported [45]. This limitation spurred research into more customizable font representations. Early attempts included Adobe's Multiple Master fonts from the year 1990, which allowed interpolation along a few design axes (such as weight or width), although they were never widely adopted in practice [16]. Another significant step was Knuth's METAFONT from the year 1979, which described glyphs via tunable parameters and mathematical equations rather than fixed outlines [31]. METAFONT enabled continuous variation and algorithmic font generation, but it struggled with representing complex letterforms and required hybrid outline strategies for intricate shapes. More recently, OpenType Variable Fonts (introduced 2016) revived the multiple-master concept in a modern form, permitting continuous variation along one or more design axes (weight, width, slant, etc.) within a single font file [37].

Parallel to these developments, new machine-learning-based approaches have emerged for automatic font creation and style transfer. For example, Hayashi et al. developed a GAN-based system (GlyphGAN) to generate stylized fonts while preserving consistency across all characters [23], and Kadner et al. demonstrated a generative font model that can adapt a typeface to maximize an individual reader's speed and comfort (AdaptiFont) [27]. In this work, font generation refers to the creation of customizable and parametric typefaces, which serve as the foundation for tailoring fonts to individual user profiles.

2.3.1 Customized Fonts

Customized fonts are typefaces that are modified or generated with user-defined parameters, rather than being static, pre-designed images. Static outline fonts require separate files for each style, and cannot smoothly interpolate among them [45]. Parametric frameworks define glyph shapes in terms of the underlying design parameters (such as stroke thickness, x-height, or slant), allowing for the automatic generation of new variants when the parameters are adjusted. For example, Hu and Hersch proposed a modular parametric font that allowed continuous variation via user-adjustable parameters [25]. Such frameworks address the "one font per style" limitation, encoding the entire style continuum within a single description. The benefits of these systems include optical scaling, a typographic principle from metal fonts that adjusts the glyph features for legibility at different sizes, reintroduced programmatically via parametric methods. These capabilities are essential for modern accessibility and personalization goals; studies indicate users read faster and feel more comfortable with fonts tailored to their preferences or needs [50]. Thus, customizable font generation combines typography, HCI, and design to create on-demand, user-optimized typefaces.

In addition to parametric and generative approaches, another avenue to achieve personalized typography is converting handwriting samples directly into digital fonts. Tools such as Calligraphr [12] allow users to digitize their handwriting by scanning predefined letter templates, creating digital typefaces that closely replicate their written script. Recent research, such as the InkSight project by Mitrevski et al., leverages advanced machine learning to convert scanned handwritten pages into digital vector strokes, effectively bridging offline handwriting and digital typography [38]. These handwriting-to-font systems capture personal form and visual identity explicitly, complementing approaches that instead focus on generating typography from behavioural or dynamic data.

2.3.2 Metaflop

Metaflop is a contemporary tool for parametric font generation, built on Knuth's METAFONT system. Developed by Marco Müller and Alexis Reigel, it is a web-based platform that enables users to adjust sliders for various glyph parameters (e.g., stroke width, ascender height) and instantly preview the resulting font [40]. In the backend, Metaflop runs a METAFONT engine that recompiles glyph outlines in real time as parameters change, and exports the resulting font in standard formats such as OTF or WOFF [40]. It includes features like randomization ("flop it"), undo, and reset, making exploration easy for non-technical users. By combining algorithmic font description with outline production, Metaflop modernizes legacy parametric capabilities through a user-friendly interface and web-based delivery, demonstrating a proof of concept for accessible font customization tools relevant to this thesis.

2.3.3 OpenType Fonts

OpenType is the current industry standard for scalable, cross-platform digital fonts, co-developed by Microsoft and Adobe in the 1990s to unify the TrueType and PostScript Type 1 formats [37]. It introduced Unicode-based encoding, advanced layout features, and broad glyph support in a single font file compatible across Windows, macOS, and other platforms.

Moreover, OpenType supports font parameterization via Variable Fonts, formalized in OpenType 1.8. A Variable Font encapsulates a continuous design space (e.g., weight, width, slant) within a single file, allowing attributes to be programmatically adjusted at runtime [37]. This enables dynamic font generation without the need for separate files for each style variant.

These capabilities make OpenType ideal for personalization systems. Parameters such as stroke thickness, x-height, and letter width can be mapped directly to user-specific inputs (e.g., typing dynamics), allowing for real-time visual customization while maintaining compatibility with existing applications. In this thesis, OpenType's flexibility is leveraged to generate personalized fonts driven by keystroke-derived features, producing standard .otf files for seamless integration into the user's environment.

The OpenType 1.8 specification defines font variations in terms of multiple interpolation axes spanning a continuous design space. For example, a single variable font may incorporate weight, width, slant, or other stylistic axes, each defined by a numeric range of values [37]. By selecting coordinates along these axes, tens of thousands of distinct font instances can be derived from one file, and the font engine renders intermediate designs at runtime. Common axes like weight (wght), width (wdth), slant (slnt), and optical size (opsz) are standardized, while type designers can also define custom axes for novel variations. This flexible architecture has made variable fonts a focal point in recent HCI and design research, as it enables tailoring typography to users and contexts on the fly. For instance, Palmén et al. adjusted a font's stroke thickness via a dedicated grade axis to improve readability across light and dark modes [46]. Wallace et al. demonstrated that matching individuals with their optimal font (or font variant) can increase reading speed by up to 35% without impairing comprehension [57]. Additionally, machine-learning approaches have been applied to recommend or even generate personalized typefaces; for example, Cai et al. presented a system that uses a font variation model to suggest reader-specific fonts, yielding significant gains in reading speed for hundreds of participants [11]. These findings underscore how OpenType's evolution, from static formats to fluid variable fonts, directly supports behaviour-driven personalization in digital typography.

2.4 Coefficient of Variation (CV)

The coefficient of variation (CV) is a standardized measure of dispersion defined as the ratio of the standard deviation σ to the mean μ of a distribution or data set. In formula form, the coefficient of variation is given by:

$$\mathrm{CV} = \frac{\sigma}{\mu} \tag{2.1}$$

Because it is a ratio, the CV is dimensionless; it is often expressed as a percentage, sometimes referred to as the relative standard deviation (RSD). This characteristic means that, in contrast to an absolute measure like standard deviation (which has the same units as the data), the CV describes variability relative to the scale of the measurements, allowing comparisons of variability across datasets that have different units or vastly different means [18]. For example, a CV of 5% indicates that the standard deviation is 5% of the mean, providing a sense of how large the variability is in comparison to the typical value.

2.4.1 Applications and Interpretation

The CV finds applications across many fields due to its scale-invariant property. In finance and investment analysis, it is commonly used to evaluate the level of risk per unit of expected return, essentially comparing the volatility of asset returns relative to their mean return as an indicator of risk-adjusted performance. In ecology and environmental sciences, the CV is employed to compare variability in populations or measurements across species and conditions; for instance, ecologists may report the CV of population counts or trait values to assess relative fluctuation magnitude regardless of absolute size. In measurement science and analytical chemistry, the CV is widely reported as a gauge of precision and repeatability for assays and instruments. A low CV in a laboratory assay implies that repeated measurements yield low variability relative to the mean result, indicating high precision [47]. Conversely, a higher CV would signal that the data are more dispersed around the mean, implying greater relative variability.

In interpreting the coefficient of variation, smaller values indicate less variability in relation to the mean (more homogeneous or consistent data), while larger values indicate greater relative dispersion. For example, $CV = 0$ would denote no variability at all (all observations equal to the mean), $CV = 0.1$ (10%) might indicate a relatively tight clustering of values around the mean, and $CV = 1$ (100%) signifies that the standard deviation equals the mean (significant dispersion relative to the average). It is important to note, however, that the CV should be used and interpreted only in contexts where the mean is defined on a ratio scale with a meaningful zero [51]. In other words, the

CV is most appropriate for data that are strictly non-negative and where zero represents the complete absence of the quantity being measured (such as lengths, concentrations, or financial returns). On an interval scale without a true zero (such as temperature in Celsius or Fahrenheit), the CV becomes arbitrary because shifting the origin (zero point) changes the ratio of standard deviation to mean [51]. Likewise, if the mean of a dataset is extremely close to zero, the CV will approach infinity, losing its interpretability; therefore, caution is required in such cases.

2.4.2 Advantages and limitations

The primary advantage of the coefficient of variation is its ability to compare variability across different datasets in a normalized way. Since it has no units, the CV allows an "apples-to-apples" comparison of dispersion for variables that may be on entirely different scales or units. This makes it invaluable in fields like finance or ecology, where one might want to contrast the variability of markedly different metrics (e.g., comparing the volatility of stock returns to the variability of growth rates in populations). Another benefit is that the CV remains constant under proportional scaling of the data. If every value in a dataset is multiplied by a positive constant, the standard deviation and mean are both scaled by that factor, leaving σ/μ unchanged. This scale-independence reflects the CV's focus on relative variability rather than absolute magnitudes [18].

However, the coefficient of variation also has important limitations. As noted, it is only well-defined for ratio-scale data with a nonzero mean; if $\mu = 0$ or if the data can take negative values around zero, the CV either diverges or becomes ill-behaved. In practice, this means the CV is not suitable for variables that can assume zero or negative values on an arbitrary scale (for example, differences or temperature in Celsius) unless those values are first transformed to a ratio scale (such as converting temperature to Kelvin). Furthermore, the CV can be sensitive to extreme values or outliers, since it relies on the standard deviation (which itself is non-robust to outliers). A single extreme observation can inflate σ and thus the CV, potentially exaggerating the impression of variability. Finally, while the CV conveys relative dispersion, it does not represent the absolute scale of variation; two datasets might have the same CV, but if their means differ by orders of magnitude, the practical significance of their variability could be very different. For these reasons, the CV should be applied judiciously, with an understanding of its assumptions and context. When used appropriately, it serves as a concise summary of variability relative to scale, complementing other statistical descriptors in a comprehensive analysis [47].

2.5 Standardized Mean Difference (SMD)

The standardized mean difference (SMD) is a statistical measure used to quantify the difference between two group means in standardized units of measurement. Essentially, it expresses how far apart two group means are, not in the original measurement units, but in units of standard deviation. The most common form of SMD is Cohen's d, defined as:

$$d = \frac{\bar{X}_1 - \bar{X}_2}{s_{\text{pooled}}} \tag{2.2}$$

where $\bar{X}_1$ and $\bar{X}_2$ are the sample means of the two groups being compared, and s_{pooled} is the pooled standard deviation of the two groups, calculated as:

$$s_{\text{pooled}} = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \tag{2.3}$$

The pooled standard deviation is a composite measure of variability that assumes the two groups have a common variance, providing a weighted average of the within-group standard deviations. By dividing the raw mean difference by the standard deviation, the SMD produces a unitless index of effect size. This allows researchers to compare group differences across studies or variables that may be measured on different scales or in different units of measurement [6].

An SMD of $d = 1$, for example, indicates that the two group means differ by one full standard deviation, which is a substantial difference regardless of the specific units involved. An SMD of $d = 0$ implies no difference between the group means. Negative values of d simply indicate directionality (e.g., if $\bar{X}_1 < \bar{X}_2$, d will be negative), though often the magnitude or absolute value is the primary focus when reporting effect size.

2.5.1 Interpretation and Practical Use

Statistically, the SMD belongs to the family of effect size measures. It provides a standardized quantification of the difference between groups, which is especially useful when the measurements are on arbitrary or incommensurable scales [14]. In practical terms, the SMD makes it possible to say "Group A is, on average, 0.5 standard deviations higher on outcome Y than Group B," which gives a sense of the magnitude of the difference in a way that is comparable across different contexts. The concept was formalized by Cohen, who introduced conventional benchmarks for interpreting the magnitude of d in the social and behavioural sciences [14]. These benchmarks (often cited as around 0.2 for a "small" effect, 0.5 for a "medium" effect, and 0.8 for a "large" effect) are not absolute rules, but they provide a typical frame of reference for discussing effect sizes. They imply,

for example, that an SMD of 0.8 is a large difference, roughly meaning that the mean of one group is 0.8 standard deviations higher than the other, which in a normal distribution corresponds to considerable non-overlap between the two groups' distributions. It is essential to note, however, that what constitutes a "large" or "small" effect can vary depending on the specific field or outcome measure [14]. In some research areas, an SMD of 0.5 may be significant, whereas in others (e.g., specific medical outcomes), even a 0.5 difference could be considered modest; thus, context should inform the interpretation of these generic benchmarks.

2.5.2 SMD in Meta-Analysis and Research Synthesis

The standardized mean difference is widely used in research syntheses and meta-analyses, where it serves as a common currency for pooling results from studies that may have used different scales. For example, one clinical trial might measure an outcome on a 20-point symptom scale, while another trial measures the same underlying construct with a 100-point questionnaire. By converting both results to an SMD, meta-analysts can combine the findings despite the different instruments used [6]. The SMD from each study (often Cohen's d or a variant thereof) can be weighted and averaged to produce an overall meta-analytic effect size. In computing an SMD for each study, one must decide on the appropriate standard deviation to use: a common choice is the pooled standard deviation of the two groups (as in Cohen's d above). If the sample sizes are small, Cohen's d tends to be a slight overestimate of the actual effect size, so a small-sample correction is applied to yield what is known as Hedges' g [24]. Hedges' g is essentially a nearly unbiased version of d, correcting for the positive bias that arises when n is small. Another variant is Glass's Δ, which uses only the standard deviation of the control (or comparison) group in the denominator; this approach is sometimes chosen if one believes the experimental manipulation might affect the variability of the outcome in the treatment group, and so prefers to standardize using the baseline variability [20]. All of these are forms of the standardized mean difference, and in a meta-analysis context, they are mathematically convertible into one another, used to summarize the magnitude of effects across diverse measurements. The use of SMD in meta-analysis is recommended when outcomes are continuous but measured on different scales, ensuring that each study contributes a commensurate effect size to the pooled analysis [6].

In addition to its central role in meta-analysis, SMD is also commonly reported in primary research as a way to convey the practical significance of results. By reporting, say, that an intervention improved test scores by "0.6 standard deviations," authors give readers a sense of impact that transcends the specific test used. This can facilitate the comparison of results across studies or contexts. However, one must be cautious when interpreting SMD values. While they convey the size of differences in standardized terms,

they do not inherently convey whether a difference is practically or clinically important. A very small SMD could be meaningful if the outcome is critical (e.g. survival time), whereas a larger SMD might be trivial for something of minor importance. Moreover, SMD assumes (when derived as Cohen's d) that the underlying variability is comparable between groups. If group variances differ greatly or if distributions are highly skewed, the meaning of a "standard deviation" unit becomes less clear. Additionally, because SMD is unitless, it can sometimes obscure the real-world meaning of a difference; stakeholders might find it easier to interpret raw differences (e.g., "a 5 kg weight loss") than an abstract number of standard deviations. For these reasons, SMD should be complemented with other information, such as confidence intervals and, where possible, raw mean differences or percentage changes, to give a complete picture of the results. Nonetheless, as a succinct index of effect magnitude, the standardized mean difference is an indispensable tool in statistical analysis, allowing comparisons and syntheses that would otherwise be impossible [14]. The conventions and statistical foundations established by Cohen and later refined by Hedges and others ensure that the SMD remains a rigorously defined and interpretable metric for differences across a wide array of scientific disciplines.

Chapter 3

Methodology

This chapter outlines the methodologies and principles that guided the design and development of the system, emphasizing the universal approaches applied throughout the process. The purpose of this chapter is to describe the abstract rules, steps, and instructions that influenced the overall design strategy. The methodologies discussed in this section include data-driven design, user-centred design (UCD), and behavioural mapping, which are the foundation for ensuring the system meets its goal of personal expression through typing behaviour analysis.

3.1 Principles of Design

The central principle guiding the design of this system is a commitment to improving individual expression opportunities. This principle aims to facilitate the creation of personalized outputs that enable users to express their unique identity through digital media. In the context of this thesis, personal expression is defined as a user's ability to generate a personalized font based on their typing behaviour, which is a direct representation of their individual cognitive and behavioural traits.

The principle of individual expression guided every methodological decision, ensuring that each choice in the system design allowed for flexibility and adaptability. This principle was used to evaluate the appropriateness of different methods and to ensure that the final system would produce outputs that felt authentically personal for each participant. By emphasizing this principle, the design remained aligned with the goal of creating a system that is not just functional but also meaningful, making the process of font generation a form of self-expression.

3.2 Methods

This section provides an overview of the universal methodologies that were used in the system design. These methods were chosen to ensure that the system would achieve its intended outcomes, i.e. the creation of personalized, user-driven fonts based on behavioural data.

3.2.1 Data-Driven Design

Data-driven design is an approach that leverages empirical data and analytics as a core input to the design process, using evidence from real-world usage to inform and validate design decisions. Rather than relying solely on intuition or anecdotal observations, designers in this paradigm collect and analyze behavioural data, usage metrics, and other quantitative insights to guide the creation or refinement of a product or system. This method has gained prominence with the proliferation of big data and IoT (Internet of Things) technologies, which provide abundant user and sensor information that can be translated into actionable design improvements [33]. Researchers have noted that data-driven design represents a major opportunity to enhance innovation and decision-making by exploiting the increased availability of user data and reducing barriers to data collection in the design cycle [30].

In practice, this means design hypotheses are continuously tested against large datasets, and patterns or anomalies in user behaviour are mapped to design features. This is a particularly valuable strategy for systems that generate personalized outputs based on user behaviour, as it ensures the design is grounded in actual usage patterns and can adapt dynamically to emerging trends revealed by the data.

3.2.2 User-Centred Design (UCD)

User-Centred Design (UCD) is a comprehensive framework that prioritizes the needs, characteristics, and feedback of end-users at every stage of the design process. The core idea is to ensure that the product or service aligns with how real users think and behave, rather than expecting users to change their habits to accommodate the design. A UCD process typically involves understanding the users' context and requirements through research (e.g., interviews, observations), iterative prototyping, and usability testing with users to gather empirical feedback. The classical principles of UCD emphasize an early focus on users and their tasks, empirical measurement of product usage, and iterative refinement of the design based on user feedback [21]. By continually centring the design process around the user's perspective, UCD helps create solutions that are intuitive, accessible, and effective at solving users' actual problems. Even in systems driven by data or algorithms, applying UCD ensures that personalized outputs and interactions are

ultimately tailored to human expectations and usability, thereby improving acceptance and the overall user experience.

3.2.3 behavioural Mapping and Affordance Theory

Behavioural mapping is an observational research method that systematically records and analyzes users' behaviours within a specific environment or system over time. The technique involves tracking where, when and how people engage in activities within a physical or digital space, and then visualizing these observations in the form of a 'map' or annotated diagram that highlights patterns of behaviour. By capturing data on user movements, actions, or choices in context, behavioural mapping allows designers and researchers to identify hotspots of frequent activity, bottlenecks, typical user pathways, and areas of disengagement. Such a map of actual behaviour can reveal mismatches between expected and actual user actions, informing design adjustments to better accommodate natural usage patterns. Designers have utilized behavioural mapping to uncover insights into how environmental factors or interface layouts influence user behaviour, thereby guiding more user-informed design decisions [42]. For a system that generates personalized digital content based on user behaviour, behavioural mapping can be instrumental in understanding the typical user journey and context of use, ensuring that personalized features are aligned with observed usage scenarios.

3.2.4 Heuristic Evaluation

Heuristic evaluation is a usability inspection technique in which expert reviewers examine a design or interface against a set of established heuristics or best-practice principles for efficient usability. Pioneered in the context of "discount" usability engineering, this method involves having a small number of experts independently walk through the interface and identify any usability issues or design shortcomings by checking compliance with recognized guidelines (e.g. visibility of system status, error prevention, consistency, etc.). [44]. Because it relies on expert judgment rather than end-user testing, heuristic evaluations can be conducted early and often, even on prototypes or design specifications, to catch obvious problems without the time and logistics needed for user studies.

By employing heuristic evaluation, designers can quickly gather a list of potential usability improvements and prioritize them before user-facing deployment. This method is particularly valuable when designing systems that involve complex interactions or personalized content, as experts can foresee where users might become confused or where the system might violate fundamental usability principles. For instance, in a personalized digital content system, an expert might ensure that recommendations are presented transparently and predictably (to avoid confusing the user) or that the interface remains consistent even as content personalization occurs, all without requiring actual end-users

to be involved in the review. Although heuristic evaluation does not provide the depth of insight that honest user feedback would provide, it serves as a cost-effective filter to ensure the design aligns with general human factor guidelines and known cognitive limitations. By addressing issues identified through expert review, the design team can improve the user experience from the outset of the project. This makes subsequent data-driven or user-centred design efforts more effective, focusing on deeper issues rather than surface-level usability problems.

Chapter 4

Data Sources

This chapter introduces the datasets relevant to this study on personalized font generation based on keystroke dynamics. We draw on benchmark keystroke datasets from the University at Buffalo, Clarkson University, and Politehnica University of Timișoara. The composition and collection methodology of each dataset are described individually, based on their respective publications and documentation. Following this, we clarify which datasets were included in our research. The Clarkson dataset, although reviewed, was excluded entirely due to its small sample size and the inconsistent availability of high-quality, usable free-text data required for our system.

4.1 University at Buffalo Dataset

Sun et al. present a comprehensive keystroke dynamics dataset collected at the University at Buffalo, containing data from 148 participants [52]. Each participant performed typing tasks in three separate sessions in a laboratory setting, with each session lasting approximately 50 minutes and separated by an average of 28 days. The keystrokes were recorded with high-resolution timing (around 15 ms timestamp precision), and the dataset was explicitly designed to include hardware variability. Four different keyboard types (Lenovo laptop keyboard, HP wireless keyboard, Microsoft desktop keyboard, and Apple Bluetooth keyboard) were used across sessions. The dataset collectors split the 148 participants into two groups: a baseline subset of 75 participants who used the same keyboard model in all three sessions, and a rotation subset of 73 participants who used three different keyboard types across their sessions. This design ensures that the dataset captures both consistent typing conditions and scenarios with keyboard variation, which is valuable for evaluating the robustness of authentication.

In each session, participants performed two main tasks defined in the Buffalo data collection protocol:

Task 0 Transcription of fixed text. Subjects transcribed a provided reference text (the 2005 Stanford commencement speech by Steve Jobs), split into three equal parts. Each session involved typing one part of the speech, so each participant transcribed the passage over three sessions. This task captures keystroke timing in a constrained copy-typing scenario.

Task 1 Free-text responses and daily activities. This task elicited natural typing behaviour through two subtasks: A) Participants answered two open-ended survey questions and wrote a description of a given image (a crowded scene with human activity), prompting free-form text input; B) Participants carried out routine computer activities mimicking a daily work session, such as logging into an email account (with provided credentials), composing a brief email, attaching the text files they had written in Subtask A, and sending the email. They could also perform brief web browsing. This free-text task generates unconstrained typing data under realistic conditions, complementing the fixed-text data from Task 0.

After data collection, all keystrokes were logged, including key event labels (key press or release) and precise timestamps (in milliseconds), for subsequent analysis. The Buffalo dataset also provides demographic information (e.g., the participants were primarily young adults, comprising 113 males and 35 females, aged approximately 20–30 years). However, demographic factors are not directly used in our feature processing.

Our study focuses on the free-text portion of the Buffalo dataset. In particular, we use Task 1 (the open-ended questions and routine email tasks) from Session 1 for each participant. We exclude the transcribed text (Task 0) and the later sessions, narrowing our analysis to each subject's natural, unconstrained typing behaviour in a single-session scenario. No data was removed beyond session and task selection, and all files associated with Task 1 were retained.

This dataset was initially collected to support research in continuous user authentication using keystroke biometrics. It reflects temporal typing variations and perturbations due to hardware changes. The dataset is publicly accessible and was used in our study under an official signed agreement with the authors (see Appendix A).

4.2 Clarkson University Dataset

The Clarkson University keystroke dataset, described by Vural et al. [56], contains typing data from 39 subjects collected in a controlled setting. Each subject attended two lab sessions, each approximately one hour in length, typically on different days. For most participants, the gap between the sessions was one to two months, introducing temporal variation in typing behaviour. Out of 39 participants, only 34 completed both sessions, with a few only having partial or single-session data. Each user's data is organized by

session, and keystroke timing has millisecond precision.

The Clarkson dataset also includes synchronized video recordings of each session, capturing the subject's face and hand movements during typing [56]. This auxiliary video data provides context (e.g., whether a participant is a touch-typist or looking at the keyboard), though our work utilizes only keystroke timing data.

During each Clarkson session, participants performed a series of typing tasks that included fixed and free-text input, as outlined in Section 2 of their paper [56]:

Password entries Each participant typed a set of three predefined passwords, repeated 20 times each within the session. These were short, fixed strings designed to collect multiple samples of precise, repetitive typing for authentication evaluation.

Free-text responses Participants answered ten open-ended questions presented in a survey format. This included eight general questions (e.g., opinions or personal prompts) and two questions asking for a description of a given image or scene. Each answer was an open-ended response typed in free text. On average, participants typed 500 characters for each question.

Text transcription Similar to the Buffalo dataset, Clarkson participants also performed a transcription task. They retyped the full text of Steve Jobs' commencement speech. The speech had two parts, with one part transcribed in Session 1 and the remainder in Session 2. This provides a fixed-text typing sample under similar conditions across the two sessions.

All keystrokes in the Clarkson dataset are timestamped and labelled by key event (down or up) and key code (a numerical identifier of the physical key). The dataset offers a mixture of structured (password, transcription) and semi-structured (free-text response) typing scenarios.

We considered using the Clarkson dataset; however, after an initial inspection, we determined it was not suitable for this research due to certain limitations that did not align with our system requirements. First, the dataset exhibits high variability in data quality and task completion across participants. Several participant sessions contain only short, fixed-text entries or password trials, while others lack complete free-text responses. As our system relies on session-level natural typing data to extract the most stable and discriminative behavioural features, this inconsistency introduces significant bias that limits the comparability of feature analysis across participants. Moreover, the dataset's small participant pool (39 participants) made it difficult to perform meaningful filtering. Filtering to retain only the few complete free-text sessions would yield a subset too small to support training or evaluation.

Given these constraints, we excluded the Clarkson dataset entirely from system development and evaluation. Nonetheless, we acknowledge it as a valuable resource for keystroke authentication research, particularly studies focused on short-text or password-based scenarios.

4.3 Politehnica University of Timișoara Dataset

Iapa and Cretu [26] published a free-text keystroke dynamics dataset collected at Politehnica University of Timișoara. This dataset comprises 80 participants, each contributing typing data during a single session via a web-based platform. All participants typed in Romanian (their native language), providing a perspective on keystroke patterns in a language with diacritics and a character distribution different from English. The frequencies of characters and digraphs differ significantly from those in the English language. Data collection was performed using a custom JavaScript key-logging tool embedded in a web form, which recorded each key event (both key presses and key releases) along with its timestamp. The timestamps had millisecond precision.

Participants were asked to provide free-text input by filling out an online form. This form included several text fields prompting the participant to type different types of content, such as answering open-ended questions or entering a passage of their choice. The content and length of the text each person entered could vary, but overall, the dataset contains a substantial amount of unconstrained typing. In total, the Timișoara dataset collected 410,633 key events across all 80 participants, corresponding to an average of approximately 5,130 key events per participant. Each participant typed a few thousand characters of free text on average. The total accumulated typing time is about 24 hours of keystroke activity spread over all participants [26]. Each recorded event entry includes at least the key identifier (key code or character), the event type (down or up), and a timestamp. This fine-grained event log enables the extraction of timing features such as key hold durations and inter-key intervals for free-text typing in Romanian.

The Timișoara dataset is inherently a single-session free-text collection, which aligns well with our requirements. We utilize the entire set of free-text keystrokes from all 80 participants and treat each participant's session as a source of authentic keystroke patterns. The whole set of participant sessions was retained without file-level exclusions, as each record met our minimum threshold for usable free-text input.

This dataset was made publicly available to advance research in keystroke dynamics and continuous authentication, particularly in underrepresented language contexts. Although we do not have a signed agreement for this dataset, the authors explicitly state their intent to support external research use in their publication. Their stated goal was to "make [the dataset] available to other interested researchers" to address the lack of public datasets in this field. We use this dataset in accordance with that declaration and

our institutional ethics approval.

4.4 Concluding Note on Permissions and Ethics Compliance

The datasets utilized in this work were obtained in accordance with institutional research ethics guidelines. Explicit signed permission was obtained for both the University at Buffalo dataset and the Clarkson dataset. The Timișoara dataset was used based on its public release and declared availability for research purposes. Full documentation of dataset permissions, agreements, and ethical compliance is provided in Appendix A. Although reviewed and authorized for use, the Clarkson dataset was excluded from this study based on the methodological limitations described above.

Chapter 5

System Design

This chapter presents the final architecture of our system, which is based on globally stable and discriminative features selected through population-level analysis and applied uniformly to new users. The system consists of four main stages, labelled A through D in Figure 5.1. In stage A, we process population-wide keystroke data aggregated from the University at Buffalo dataset and the Politehnica University of Timișoara dataset, as described in Chapter 4, to extract event attributes with high intra-participant stability and inter-participant discriminability. Features that meet both criteria are promoted to a global feature list. In stage B, these globally stable and discriminative features are combined into a consolidated, population-wide ranked list of the top n features ($n = 6$ in our application). This list is fixed and shared across all participants, ensuring consistency and interpretability. Stage C captures the typing behaviour of a new user or participant (either one of the 12 test participants or an online user). Their raw keystroke data is processed to compute only the event attributes corresponding to the global top features; no further stability evaluation is performed at this stage. Finally, stage D maps these extracted feature values to font parameters using a fixed, rank-based scheme. The font parameters are passed to the generator, resulting in a downloadable, personalized font.

5.1 Population vs. Test Data

In developing this system, we divided our collected keystroke dataset from Chapter 4 into a population dataset of 218 participants and a test dataset of 12 typists, ensuring no overlap to maintain evaluation integrity. The population dataset consists of keystroke data aggregated from a broad set of participants and serves exclusively to derive globally highly stable and discriminative features. These features are selected based on their statistical properties across the entire population, not based on any individual participant's profile.

The test dataset is completely disjoint from the population dataset and is used solely

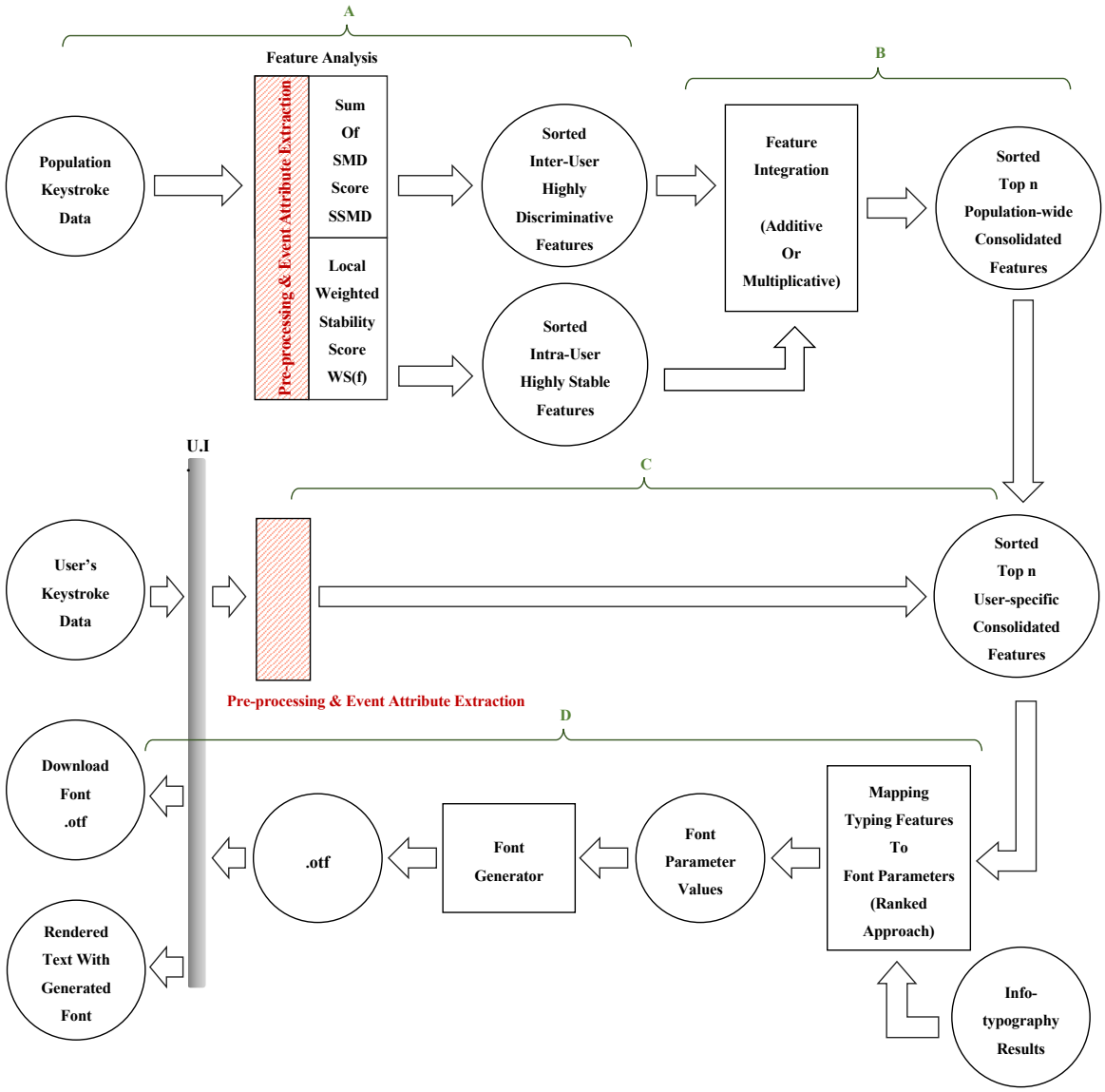

Figure 5.1: System architecture: The system derives globally high stability and discriminability features by analyzing population-wide keystroke data (Parts A & B). This consolidated, population-based feature set is fixed and applied to all participants. For each new user or participant (Parts C & D), only the values corresponding to the pre-selected features are extracted from their keystroke data and mapped to font parameters for generating the personalized font. No per-user stability evaluation influences feature selection at this stage.

to evaluate the generalizability and personalization of the fixed global feature list. In other words, no feature selection or filtering is performed on the test participants; they are evaluated strictly using the features derived from the population dataset. This separation ensures an unbiased assessment of system performance and prevents data leakage between the feature selection and evaluation phases.

The population set includes participants with sessions yielding over 11,000 feature rows after augmentation, enabling the system to identify stable features that scale across real-world scenarios. Our test set comprises 10 randomly selected participants who meet the same quality threshold, along with writing samples from the thesis author and the project supervisor, totalling 12 test participants.

5.2 Data Preprocessing and Transformation

The source datasets differ in structure and format, but each was ingested independently into the feature extraction pipeline using dataset-specific loading scripts. No data merging was performed. Instead, each participant's data was parsed, pre-processed, and normalized to a standard schema (i.e., key name, down timestamp, up timestamp). Dataset-specific structure (e.g., JSON vs. plain text logs) was accounted for during ingestion. This modular approach ensures that differences in data structure or recording environments do not confound feature extraction.

Data from raw keystroke logs was then cleaned and transformed before feature extraction using a series of preprocessing steps. The preprocessing pipeline was designed to improve data quality and prepare features for augmentation. This pipeline included outlier removal, normalization, and feature calculation, as detailed below:

Outlier Removal We began by filtering out extreme timing values in the keystroke data using the Interquartile Range (IQR) method. For each participant and each type of timing measurement (feature), we computed The first and third quartiles (Q_1 and Q_3) and removed any data points that fell outside the range $[Q_1 - 1.5 \times IQR; Q_3 + 1.5 \times IQR]$. This method is more robust for keystroke dynamics than a simple percentile cutoff because typing data often contains irregular pauses or bursts (for instance, a sudden long pause if the participant is distracted or a very short interval if the participant double-tapped a key). The IQR-based filter adapts to the spread of the middle 50% of the data, ensuring that only true anomalies are discarded. By removing these outliers, we prevent skewed or unrepresentative timing values from distorting feature calculations (e.g., an abnormally long pause would artificially inflate the average value for that participant if not removed).

Normalization and Structuring After outlier filtering, the remaining keystroke data is normalized and organized into a structured format for analysis. Normalization involves scaling certain features to a standard range or unit to make them comparable (for instance, if one feature is measured in milliseconds and another is dimensionless, we may scale or standardize the values so that no single feature dominates due to unit differences). We also structure the data by ordering events chronologically and grouping them by participant and session. This step involves computing low-level metrics from raw

timestamps, such as calculating the duration of each key press and the inter-key delay (the time between releasing one key and pressing the next). This structure yields a clean feature matrix for each participant, where each row corresponds to a specific feature of their typing and each column corresponds to an event (key-up or key-down timestamp).

Feature Computation and Transformation In this step, we transformed the cleaned and structured data into high-level statistical features that serve as inputs for both analysis and font generation. The process begins by deriving event attributes from the raw keystroke logs, such as key hold durations and inter-key latencies, which are calculated from the individual press and release events. These post-augmentation values form the rows of an intermediate data structure for each session. Next, we compute features by aggregating these event attributes over the entire session or within a defined partition. These aggregated statistics form a compact summary of a participant's typing behaviour and are used as the basis for downstream feature scoring and font generation.

Before we describe the extracted features in detail, let's clarify the structure of the data used throughout the pipeline. The data in our system operates on three hierarchical layers:

- Events: The raw keystroke logs, each consisting of a key press and/or release along with its associated timestamp.
- Event attributes: Derived values calculated from pairs or sequences of events (e.g., key hold duration, inter-key latency), representing post-augmentation keystroke characteristics. These form the rows of the intermediate data structure produced after preprocessing.
- Features: Aggregated statistical descriptors (e.g., mean, standard deviation) computed over event attributes for a specific participant session or partition. These high-level features are what the system ultimately uses to map onto font parameters.

This distinction is important to prevent confusion between the fine-grained data collected at each keystroke (event attributes) and the higher-level statistical summaries used to generate the personalized font (features).

5.3 Features Types

This section details the specific event attributes extracted from keystroke data, such as key hold durations and inter-key latencies, as well as the statistical features derived by aggregating these attributes over a session or partition. For the population dataset,

these features serve as candidates for evaluating stability and discriminability, which informs the system's final feature selection. For the selected test participants or the online users, only the event attributes corresponding to the pre-selected top features are aggregated to produce the final feature values required for font generation. For example, a participant might press keys very briefly and immediately after one another, whereas another participant might exhibit longer latencies or even negative release-to-press times under conditions of stress or distraction. There are three main types of features that the system extracts, which we discuss in the following subsections: monograph features, digraph features, and augmented features.

5.3.1 Monograph Event Attributes and Features

Monograph features reflect the duration for which a single key is held down before being released. This time interval, also referred to as dwell time, is calculated as the difference between the timestamp of the key press and the timestamp of the corresponding key release. These durations are computed as event attributes for each key occurrence in a participant's session, capturing how long a participant typically holds individual keys during typing.

For example, if the letter "e" is typed ten times in a session, the system records ten separate dwell times, each representing the duration from the press to the release of that specific instance. This process is repeated for all keys, including letter keys (e.g., a, s, t), punctuation keys (e.g., '.', ','), and special keys such as Space, Shift, and Enter. While most letter keys are typed frequently enough to produce meaningful statistics, keys like Shift or Backspace may appear less often but can still be informative for some participants.

Once these individual dwell times are recorded, the system aggregates them across a session (or partition) to produce summary statistics, such as the mean for each key. These aggregated values are what we refer to as monograph features. These features are useful because they capture consistent timing patterns in motor execution. For instance, some participants may consistently press the spacebar more firmly (resulting in longer hold times), while others may type letters rapidly with minimal key hold duration. By including a wide range of keys and aggregating their dwell characteristics, monograph features provide a detailed temporal fingerprint of how each individual interacts with their keyboard.

5.3.2 Digraph Event Attributes and Features

Digraphs represent the temporal relationships between two consecutive keystrokes. Each digraph is initially derived as a set of event attributes computed from key press and

release timings of successive keystrokes. Specifically, we derive four types of digraph timing attributes, defined as follows:

Up–Down (UD) Time between releasing one key and pressing the next. This interval, also known as latency time, reflects the duration of time the participant pauses between keystrokes.

Down–Up (DU) Time between pressing one key and releasing the next, also known as the DU flight time. This captures overlapping movement and is sensitive to the fluidity or any hesitation during typing.

Down–Down (DD) Time between pressing two consecutive keys (DD flight time). DD is often used as an indicator of overall typing momentum and rhythm.

Up–Up (UU) Time between releasing two consecutive keys (UU flight time). This measures how long both keys are off the keyboard and can indicate cognitive load or motor delay in transitions.

Like the monograph values, these digraph timing values are first recorded as individual event attribute entries (one for each occurrence of a digraph). Then, for each digraph type, aggregate statistics (e.g., the mean UD time across all occurrences of the letter pair "th") are calculated to form the final set of digraph features. These session-level summaries allow us to characterize typical inter-key behaviour and support further modelling of participant typing patterns.

Figure 5.2 provides a visual representation of the temporal measurements that shape this process. By parsing raw keystroke logs into a timeline of key press and release events, the system calculates monograph and digraph latencies as soon as it launches. These low-level measurements provide the structural foundation for constructing high-level features.

5.3.3 Augmented Features

To deepen the analysis, the system also considers:

Typing Speed Calculated over a rolling window of ten keystrokes to analyze pacing variations. This feature captures short-term fluctuations in a participant's typing tempo rather than relying on a global average. It helps isolate bursts of fast typing, moments of hesitation, or transitions between fluid and interrupted keystroke sequences. Typing speed is a compound indicator that reflects not only motor speed but also familiarity with the typed content and cognitive engagement during input.

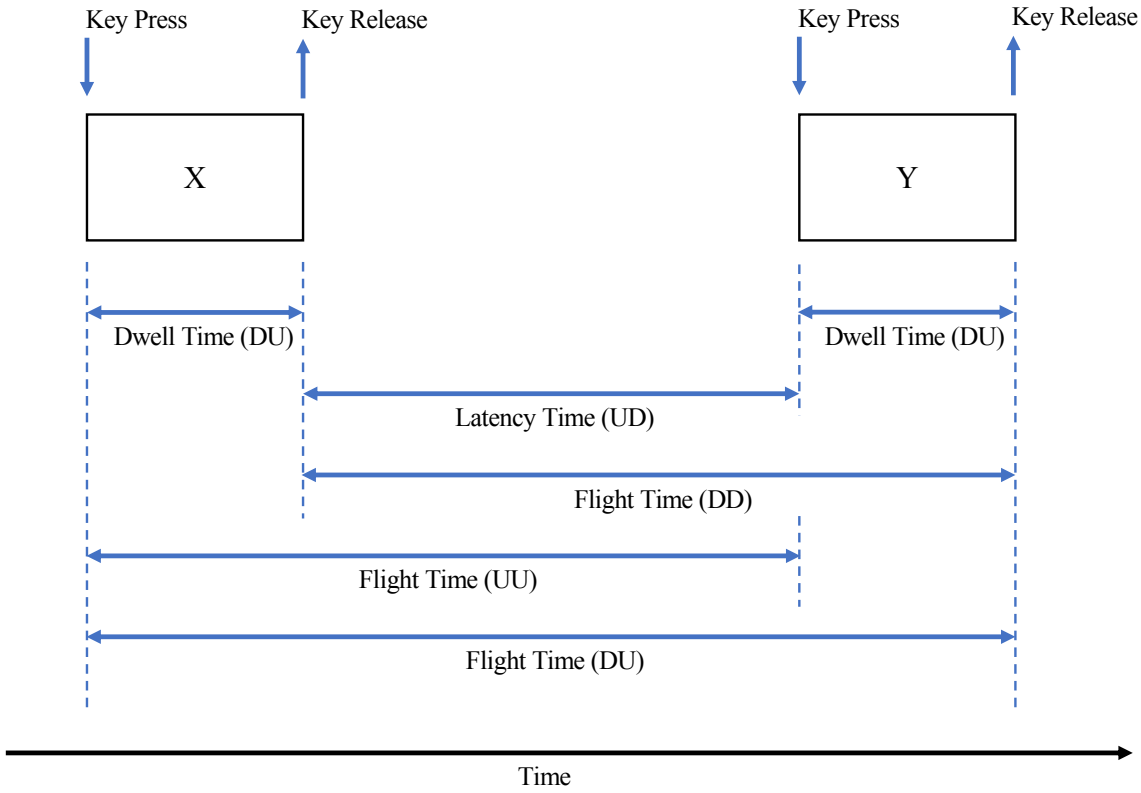

Figure 5.2: Monograph and Digraph Latencies: This figure illustrates the timing relationships between key presses and releases, including Dwell Time (DU), Latency Time (UD), and Flight Time (DD, UU, DU).

Left/Right Typing Zone Classification of keys based on the keyboard side to analyze spatial preferences. Each key is assigned to either the left or right zone according to its position on a standard QWERTY layout. This feature enables the system to detect asymmetries in hand usage, such as a dominant hand or keyboard-specific bias. Differences in zone-level timing or key utilization patterns contribute to identifying subtle biomechanical habits unique to each participant.

The features described above are essential for building a profile that accurately reflects the participant's typing signature and, ultimately, for generating a personalized font.

The following table summarizes the selected features.

These extracted features are subsequently passed into stability and discriminative analyses, which determine their suitability for representing individual typing styles and for distinguishing among participants at a population level.

Feature Type	Name	Description
Monograph (Hold Time)	M	Duration between key press and release
Digraph (Up-Down Time)	UD	Time between releasing one key and pressing the next
Digraph (Down-Up Time)	DU	Time between pressing one key and releasing the next
Digraph (Down-Down Time)	DD	Time between pressing two consecutive keys
Digraph (Up-Up Time)	UU	Time between releasing two consecutive keys
Typing Speed	Speed	Average typing speed over a window of 10 keystrokes
Spatial Typing Zone	Left/Right	Zone-based classification of keystrokes (Left or Right)

Table 5.1: Extracted features used for the typing behaviour analysis

5.4 Feature Analysis

In this section, we explore two interrelated characteristics of feature behaviour: discriminability and stability. High discriminability features are those that most effectively distinguish between different distributions or participants; for example, features with higher predictive power or a large between-class variance compared to within-class variance. These features are identified based on their correlation with the target outcome or ability to distinguish between distinct participant identities. High stability features, on the other hand, are characterized by consistent behaviour in a single participant's data. A feature is stable when it exhibits low variability for the same participant across time or sessions, indicating a reliably repeatable pattern.

Often, the most discriminatory features are not necessarily stable (they might fluctuate significantly for a given participant) or vice versa. Our analysis explicitly targets both aspects. We distinguish between intra-participant (local) high-stability features, which are consistent within each individual participant's session, and inter-participant (global) high-stability features, which remain among the top locally high-stability features for most participants in the population. A globally high stability feature is one that not only remains steady for each participant but also exhibits consistent and stable behaviour across the majority of the participant population. Identifying locally high-stability features enables personalized modelling (tuning parameters per participant), while globally high-stability features are valuable for creating one-size-fits-all models, as their patterns hold universally. Combining these features with the discriminative ones ensures that they provide meaningful differentiation and are not merely stable across most partici-

pants without adding any discriminatory characteristics.

5.4.1 Session Length Filtering

Before performing feature stability or discriminative analyses, we apply a session length filter to remove sessions that are too short. The rationale is that extremely short sessions, with very few key events, do not contain enough information to characterize feature behaviour reliably. Such sessions can distort the analysis; for instance, participants with limited data might appear to have artificially high stability (due to low variance) or produce misleading feature values due to artifacts from a limited sample. To address this, we enforce a minimum session length threshold. Only sessions containing at least 11,000 key events were retained in the population dataset. This ensures that each session can be partitioned into multiple segments (four in our case) with enough samples per segment to support robust within-session comparisons.

Figure 5.3 shows the distribution of total key events per session across all participants. The red region on the left highlights sessions that fell below the 11,000 threshold and were excluded. The remaining sessions (in blue) meet the required size and were used in our analyses of feature stability and discriminability.

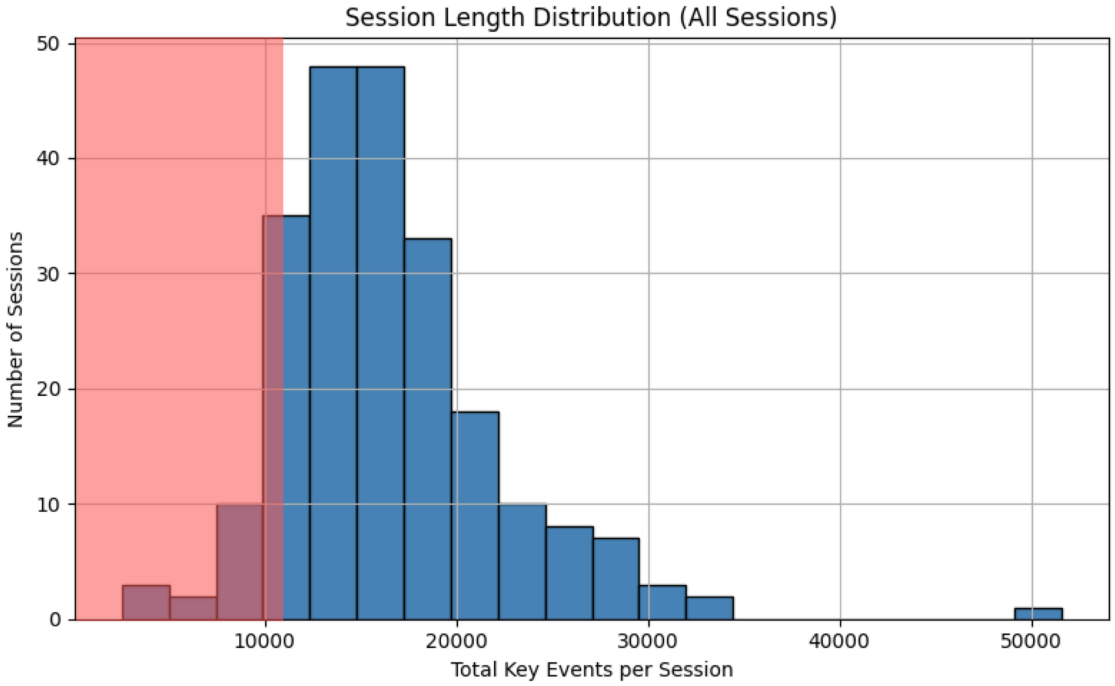

Figure 5.3: Histogram of total key events per session across all participants. The red-shaded region on the left highlights sessions with fewer than 11,000 events, which were excluded from stability and discriminability analysis due to insufficient data for partitioning. The remaining sessions (unshaded) were retained for reliable feature evaluation.

This filtering retains the sessions that contribute most to stability assessment while

discarding shorter sessions that yield too few partitions to provide meaningful stability assessment. Therefore, by filtering out undersized sessions, we enhance the reliability of subsequent analyses. Feature stability calculations are based on several observations per participant, and participants with trivial amounts of data do not skew discriminability comparisons.

5.4.2 High discriminability in features

Highly discriminable features are those that contribute most significantly to distinguishing participants. In the context of keystroke dynamics, a feature is considered discriminatory if its distribution differs considerably between participants. This differentiation can manifest itself in various ways, including different means, spreads, or overall shapes of distributions. The underlying assumption is that certain typing behaviours, such as key hold durations or specific digraph latencies, are inherently more participant-specific than others. In essence, this score compares the mean of feature f for participant u to the mean for all other participants, normalized by the variance; a higher score indicates that participant u is more easily identifiable by that feature. Features can then be ranked by their Sum of Standardized Mean Differences (SSMD) scores to find which ones are most distinctive for each participant or overall. Typically, a high SSMD means feature f takes on values for participant u that are far from the others' values and with relatively low dispersion (noise), marking f as a strong candidate for distinguishing that participant.

The intuition behind identifying high discriminability features is visualized in Figure 5.4. In the left panel, the distributions of a feature across participants overlap significantly. This feature does not help tell participants apart because their values are too similar. In the right panel, the distributions are well-separated, showing that participants have distinct values for this feature. The greater the separation, the higher the feature's discriminative power.

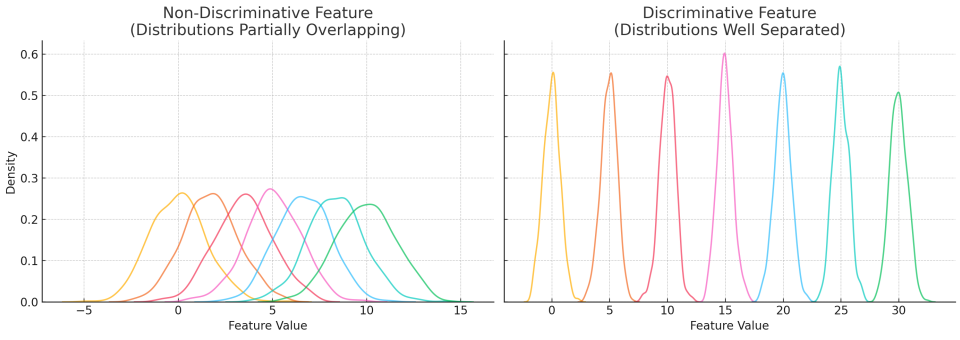

Figure 5.4: Illustration of feature distributions across participants. This is not real data; it is an idealized illustration for clarity. Left: an event attribute with low discriminability power, having overlapping distributions; Right: an event attribute with high discriminability, having well-separated distributions. This separation is the basis for our feature ranking.

It is worth noting that a high SSMD implicitly requires a low within-participant variance (for that feature), which is related to the notion of stability. This connection implies that there is some overlap between the concepts of discriminative power and stability, since stable within-participant behaviour makes it easier for a feature to differentiate reliably against that participant. However, the SSMD-based discriminative metric and our local stability metric, defined below, are not redundant but complementary to each other. SSMD captures relative distinctiveness, meaning how separated a participant's event attribute statistics are from the population, whereas local stability captures absolute consistency in the participant's behaviour irrespective of others. For example, a feature could yield a very high SSMD for a particular participant (whose average is far above the average for the rest of the population), but if that participant's values vary significantly over time, their local stability score would be low. Conversely, a feature might be highly stable to a participant (exhibiting minimal variance in their session), but if that participant's stable value lies near the population average, the SSMD will be low, as the feature does not distinguish the participant from others. Therefore, we evaluate both: SSMD ensures that a feature is characteristic or unique to the participant, while local stability (as discussed in the next section) ensures that the feature is reliably consistent for that participant. Both properties are desirable for robust participant-specific features.

5.4.2.1 Feature Discrimination Score

To quantify this discriminative ability, we employ the sum of Standardized Mean Differences (SMD) across all participants. The SMD for each pair of participants i and j for a given feature f is defined as:

$$\mathrm{SMD}_{ij}^{f} = \frac{|\mu_i^f - \mu_j^f|}{\sqrt{\frac{(\sigma_i^f)^2 + (\sigma_j^f)^2}{2}}}$$

Where μ_i^f and μ_j^f are the mean values of feature f for participants i and j, and σ_i^f, σ_j^f are their standard deviations.

The total discriminative score of a feature f across the population is the sum over all participant pairs:

$$\mathrm{SSMD}_f = \sum_{i<j} \mathrm{SMD}_{ij}^{f}$$

A high SSMD_f means that the feature shows large mean differences and low within-participant variance, i.e., it is well-separated between participants and consistent within each participant.

The ranking is derived by sorting features in descending order of their SSMD scores.

This ensures that the top-ranked features are those that most clearly delineate participant distributions.

This methodology allows us to compare all features uniformly and supports selection based on quantitative evidence of separability. In our implementation, we used this ranking to form the basis of feature selection before integrating the stability score.

For an empirical example, Figure 5.5 shows the density distributions for a representative monograph feature 'A (M)' across 10 participants. This helps visualize how distributions may or may not overlap, offering qualitative validation of the SSMD score.

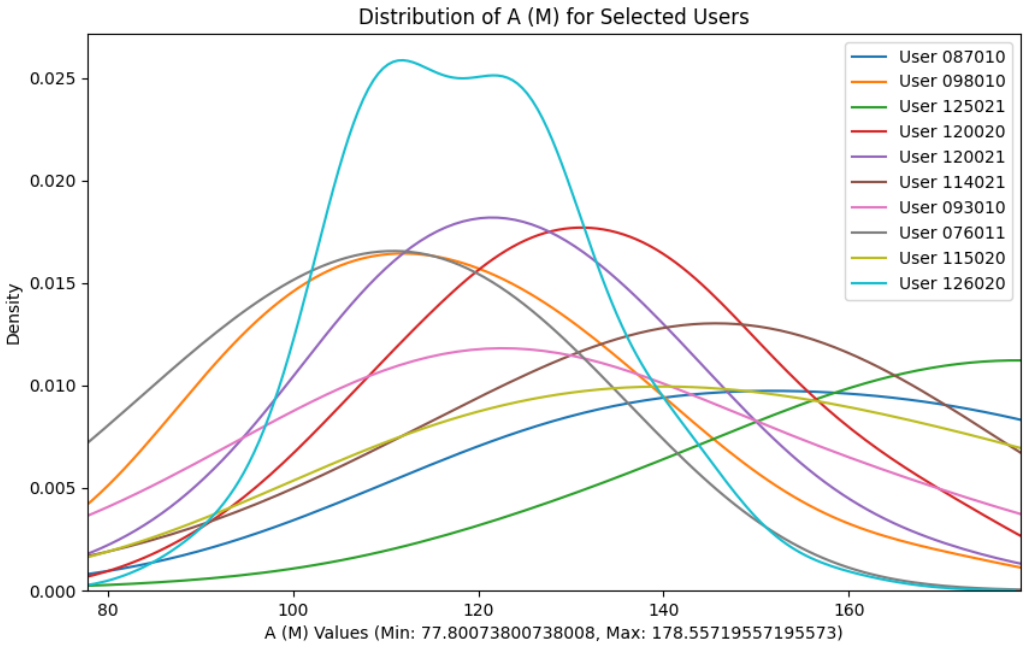

Figure 5.5: Density distributions of feature 'A (Monograph)' for ten randomly selected participants. While most distributions are centred around different values, there is a tendency for them to overlap as well. Features with less overlap will score higher on SSMD.

This analytical approach gives us confidence in the reliability of SSMD as a measure of inter-participant discriminativeness and justifies its use in subsequent stages of our feature selection pipeline.

5.4.3 Features with Stable Characteristics

In addition to being discriminative, a suitable group of selected features should exhibit consistent behaviour throughout a session. We define high stability features as a group of features that maintain relatively steady values over time (or usage) for an individual session. In other words, a high stability feature does not fluctuate wildly or change its statistical properties significantly from one part of a session to another. We assess

feature stability by examining each participant's session in partitions and measuring the variation in a feature's value across these partitions. The following subsections describe our methodology for identifying locally and globally high-stability features.

5.4.3.1 Local Stability in Features

To evaluate stability within an individual session (i.e., a single participant's data), we partition each session into multiple segments and then observe the behaviour of the features in each segment. Rather than dividing sessions according to elapsed time, which could result in uneven or inactive segments, we segment sessions by feature count; therefore, each segment contains an equal number of consecutive key events (observations). This approach ensures that each partition represents a comparable sample of the session, avoiding biases due to variable sampling rates or idle periods. We found that partitioning by equal time intervals was not suitable for our data because participant activity is not uniform over time. Some time windows contain many observations while others have fewer or none, which skews the analysis. By partitioning by event count instead, each segment provides a robust sample of feature values, and no segment is underrepresented due to low activity.

We divide each session into n equal-sized partitions (in our case, 4). For each feature f, we compute a representative statistic in each partition, typically the average value $\bar{x}_{f,i}$ of feature f in partition i. By examining the sequence of the partition means $\{\bar{x}_{f,1}, \bar{x}_{f,2}, \ldots, \bar{x}_{f,n}\}$, we can gauge how stable the feature's values remain throughout the session. If these partition means are all very similar (low variability), the feature is locally stable for that participant. Conversely, if the partition means differ substantially (high variability), the feature's behaviour is not consistent over time, indicating instability during that session.

To quantify this intuition, we define a local stability score for feature f based on the variation of its partition mean. Let $\sigma_f^{(p)}$ denote the standard deviation of $\{\bar{x}_{f,1}, \bar{x}_{f,2}, \ldots, \bar{x}_{f,n}\}$ across the n partitions of the session. We then normalize this variation relative to the feature's global range to obtain a dimensionless stability measure. Specifically, we use the effective global range $R_f^{(\text{eff})}$ of feature f, defined as the difference between high and low percentile values of f (for instance, the 5th to 95th percentile) in the entire session. This trimmed range captures the typical span of feature values while excluding extreme outliers that could distort the scale. The local stability score is formulated using an exponential decay function, so that even moderate variability will slightly lower the score, while large variability will significantly lower it. The score for the feature f in a given session is given by:

$$S_{\text{local}}(f) = \exp\left(-\frac{\sigma_f^{(p)}}{R_f^{(\text{eff})}}\right)$$

Under this definition, a feature with zero partition variability ($\sigma_f^{(p)} = 0$) achieves $S_{\text{local}}(f) = 1$, the maximum stability. As the variability $\sigma_f^{(p)}$ increases relative to the feature's range, the exponential term decays toward 0, indicating diminishing stability. Because of the exponential formulation, small fluctuations (where $\sigma_f^{(p)} \ll R_f^{(\text{eff})}$) result in a score only slightly below 1, whereas extremely large fluctuations (comparable to the feature's entire value range) drive the score down markedly. This provides a smooth, non-linear penalty for instability that never produces negative values and naturally bounds the stability score between 0 and 1.

We contrasted this approach with an alternative formulation based on the coefficient of variation (CV). A CV-based stability measure would normalize partition variability by the feature's mean, rather than its range. For example, a score could be:

$$S_{\text{CV}}(f) = \exp\left(-\frac{\sigma_f^{(p)}}{\mu_f}\right) = W_f$$

Where μ_f is the overall mean of the feature f in the session (using $|\mu_f|$ if the mean is near zero). In this scheme, stability depends on variability as a fraction of the feature's typical value. While the CV-based score also lies between 0 and 1 (and decreases with increasing variability), it introduces a strong dependence on the feature's absolute magnitude. Features with very small means (close to zero) can be harshly penalized by S_{CV}, since any non-zero $\sigma_f^{(p)}$ may be larger relative to μ_f. This can make even a fairly consistent feature appear unstable if its values cluster around zero. On the other hand, features with large means might receive a milder penalty for the same absolute variability.

Figure 5.6 compares the behaviour of the two stability score formulations. The exponential range-based score $S_{\text{local}}(f)$ (our chosen method) is plotted alongside the CV-based score $S_{\text{CV}}(f)$ for various hypothetical scenarios of partition variability. We observe that the range-based method offers a more uniform treatment across different value scales; it only considers the span of variation relative to feasible values, rather than the mean level. The CV-based method understates stability for low-mean features and is highly sensitive to μ_f. For our data, we found that $S_{\text{local}}(f)$ was more robust in ranking feature stability, particularly for features with small average values or skewed distributions. The range-based stability score gracefully handles cases where a feature's distribution includes occasional spikes or outliers: by using $R_f^{(\text{eff})}$ (excluding extreme tails), a few rare large values do not completely dominate the stability assessment.

Using the chosen stability score $S_{\text{local}}(f)$, we evaluated each feature in each eligible session. Features that consistently yield high S_{local} (close to 1) in a session are deemed locally stable for that participant, meaning the feature's behaviour is steady over the course of that session. In contrast, features with low scores in a session (significantly below 1) are unstable in that session, as their partition means vary widely. It is important to note that the skewness in the feature distribution can affect this outcome. If a feature's

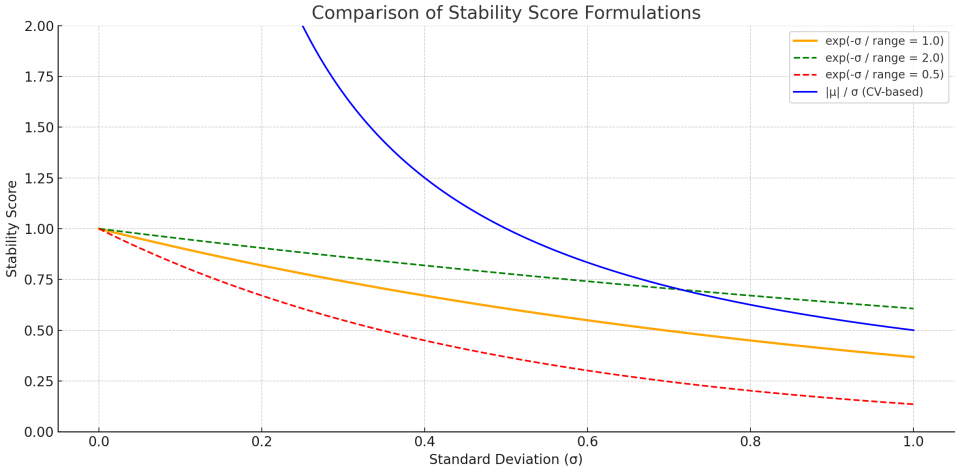

Figure 5.6: Comparison of two stability score formulations for a feature (idealized illustration using synthetic data): the proposed normalized standard deviation with exponential decay (blue curve) versus a coefficient-of-variation-derived formulation (orange curve). The exponential range-based score S_{local} penalizes variability relative to the feature's effective range, while the CV-based score S_{CV} penalizes variability relative to the feature's mean. The plot illustrates that S_{CV} can severely penalize features with low mean values, whereas S_{local} offers a more balanced assessment across different value scales.

values are highly skewed or heavy-tailed, one partition might contain an outlier or a burst of high values that substantially increases $\sigma_f^{(p)}$.

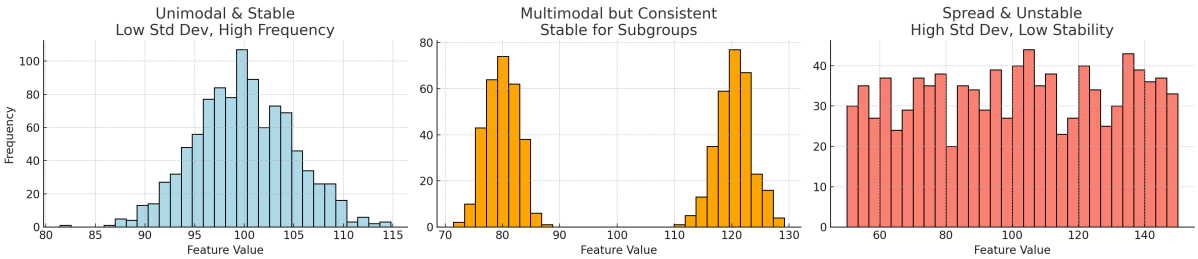

Figure 5.7: Example of a feature that exhibits a wide spread of values within a session, with a heavy skew (long tail). This is based on synthetic data and serves as an idealized illustration for clarity. One partition includes several extreme values (outliers), resulting in a high $\sigma_f^{(p)}$ and consequently a low stability score S_{local}. Even after excluding outliers beyond the effective range, the feature's inherent variability qualifies it as unstable.

By applying the above procedure, we identify locally high stability features as those features that achieve consistently high $S_{\text{local}}(f)$ in the long sessions. These are features that each participant can rely on during a session without experiencing significant drift or fluctuation in their values. Locally high stability features are desirable for modelling because they provide a steady signal over time, thereby reducing the likelihood of temporal variations misleading the model.

5.4.3.2 Global Stability in Features

While $W_u(f)$ quantifies how stable a feature f is within a single session of participant u, our ultimate goal is to identify features that exhibit stable behaviour across the population. It is worth noting that each participant in the population has only one typing session, which means local stability always refers to within-session consistency, not across sessions.

To aggregate these local insights and determine whether a feature is globally stable, we compute the mean of $W_u(f)$ across all participants:

$$GlobalStability(f) = \frac{1}{N} \sum_{u=1}^{N} W_u(f) \tag{5.1}$$

Here, N is the total number of participants (i.e., sessions), and each $W_u(f)$ is in the interval $[0, 1]$. This produces a global stability score that reflects how consistently features f maintain low intra-session variability across the participant base.

A feature will score high in $GlobalStability(f)$ if many participants express it in high $W_u(f)$, indicating frequent and consistent expression. Conversely, if f is unstable for most participants or rarely appears, the score is low. Importantly, this global measure is not a strict filter; we do not threshold $W_u(f)$ values (e.g., discard anything below 0.5). Instead, we use the mean value, which allows partial values to contribute proportionally to the overall result. For instance, if 50% of participants exhibit a high $W_u(f) \approx 1$ and the rest have $W_u(f) \approx 0$, then $GlobalStability(f) \approx 0.5$, reflecting moderate consistency. This soft aggregation avoids over-penalizing features that are selectively stable.

5.4.3.3 Distinction between Global and Local Stability

The difference between the application of local and global stability is demonstrated in our dual-axis system. We compute $W_u(f)$ using only the test participant's data, as reflected in axis A of our original architecture (Figure 5.1). This is a purely local stability check, which evaluates the reliability of features for that specific individual without making population-wide generalizations. This approach ensures that the features are specifically reliable for the test participant's session, as we are only concerned with their specific typing behaviour and not how those features behave across the entire population.

In contrast, the redesigned architecture in Figure 5.1 (Axis A) utilizes only the population data to compute $W_u(f)$ for each feature-participant pair. The subsequent global aggregation produces $GlobalStability(f)$. This version of the pipeline does not verify the stability of the incoming user or test participant's session. Instead, it filters features based on their already-established reputation for consistency across the training (population) set. This shift implies that when the top flow is used, feature selection relies on population statistics that have the potential to generalize to unseen users, thereby reducing

computational overhead and making runtime evaluation more efficient. Importantly, this process skips per-session feature analysis of the incoming user data, instead relying on pre-calculated, globally high-stability and discriminability features from the population dataset. These features, once selected, are assumed to be stable across the population and suitable for new, unseen users.

In addition to system optimization, this separation reflects a conceptual distinction in goals:

- Local stability prioritizes session-specific reliability, ensuring that features are consistent for the individual participant's typing behaviour.
- Global stability emphasizes cross-participant robustness, ensuring that features are reliable across an entire population and providing a chance for the statistics to be extended to unseen users in subsequent sessions.

To summarize, while both architectures aim to identify features with high local stability, they diverge in their methodology. In the original design (Figure 5.1, Axis C), stability is computed in real-time using only the current participant's or user's data, ensuring session-specific reliability. Conversely, the expanded design (Figure 5.1, Axis A) bypasses this per-participant computation by leveraging globally pre-evaluated stability metrics from population data (Parts A & B), integrating them with discriminability scores for consolidated feature selection. This population-informed pathway prioritizes efficiency and generalization, assuming globally robust features remain valid across unseen users without session-specific validation.

5.5 Integrated Feature Set

Having assessed the discriminative power and stability of each feature, we construct an integrated feature selection metric that combines both criteria to evaluate the overall performance. The aim is to select features that are not only informative for distinguishing classes but also consistently reliable over time. We explore two combination strategies for merging discriminative and stability scores: an additive (weighted sum) approach and a multiplicative approach. After obtaining an integrated score for each feature, we determine the final ranked feature set and ensure the ordering reflects our primary discrimination goal, as described below.

5.5.1 Additive Combination

In the additive approach, we compute a weighted sum of the normalized discriminative and stability scores for each feature. The scores are already normalized to the range of

[0, 1]. Let D_f be the normalized discriminative score for the feature f and T_f be the normalized stability score (we use T_f to denote stability to avoid confusion with the letter S for the score). We then define the combined score as:

$$C_f^{(\text{add})} = w\, D_f + (1 - w)\, T_f$$

Where $0 \leq w \leq 1$ is a weight that controls the emphasis on discriminative power vs stability. A higher w prioritizes discriminative ability, while a lower w prioritizes stability. In the special case of $w = 0.5$, both criteria contribute equally. For our analysis, we chose $w = 0.5$ to treat discrimination and stability as equally important objectives. In the absence of a reason to favour one over the other, a balanced weighting seemed appropriate.

The additive combination produces a single score $C_f^{(\text{add})}$ for each feature. Features can then be ranked by their score, and those with high discriminative power and high stability will naturally rise to the top. Features that are excellent in one aspect but weak in the other will receive intermediate scores. For example, a feature with outstanding discriminative ability but only mediocre stability might still rank well, albeit slightly below a feature that excels in both aspects. Conversely, a feature that is extremely stable but only modestly discriminative could outrank a feature that is very discriminative but highly unstable, depending on the weight chosen w. The additive method thus allows for a trade-off: a deficiency in one criterion can be compensated for to a certain extent by excellence in the other. This flexibility can be useful if we suspect that some slightly unstable features might still be worth including due to their high class-separation power or vice versa.

5.5.2 Multiplicative Combination

The multiplicative combination approach takes a strict stance: A feature will score high only if it excels in both discriminative power and stability simultaneously. In this scheme, we calculate the combined score as the product of the two normalized scores:

$$C_f^{(\text{mult})} = D_f \cdot T_f$$

Both D_f and T_f are in the range $[0, 1]$. (One could generalize to $C_f = D_f^{\alpha}\, T_f^{\beta}$, but for simplicity and equal weighting, we can use the direct product, effectively $\alpha = \beta = 1$.) The key characteristic of the multiplicative scores is that if either D_f or T_f is low, the product will be low. There is no linear trade-off as in the additive case; instead, the criteria reinforce each other. For instance, if a feature has $D_f = 0.9$ (very high discriminative) but $T_f = 0.3$ (low stability), its multiplicative score is 0.27, which is relatively low; this feature would likely be dropped from the top ranks. In contrast, if another feature

had $D_f = 0.8$ and $T_f = 0.8$, its product would be 0.64, reflecting a more balanced strength in both dimensions, and it would be ranked higher than the previous feature. This behaviour penalizes imbalance: any feature that is disproportionately strong in one aspect but deficient in the other will not score as well as well-rounded features.

The multiplicative combination is useful when we want to ensure that selected features have no major weaknesses. It effectively filters out features that, while promising in one regard, might pose problems in another (for example, a high discriminability feature that fluctuates too much during a session could confuse a model deployed in a real-time setting, so the multiplicative score naturally downgrades it). However, this strictness can also be a drawback. It might eliminate some features that are borderline in one metric but outstanding in the other, which an additive approach might retain for consideration. Therefore, in practice, we evaluate both $C_f^{(\text{add})}$ and $C_f^{(\text{mult})}$ rankings to see which strategy offers the most comprehensive set of features for our task.

5.5.3 Re-ordering the Features

After computing the integrated score, we need to produce the final ordered list of selected features. We decide on a cutoff or a specified number of top features to retain. In our application, we specifically extract the top 6 features, and the reason for this choice will be discussed in Section 5.8, Font Parameters. In short, this corresponds to the six font parameters that are subsequently modified to generate personalized fonts.

One straightforward way to finalize the feature list would be to take the sorted list directly by $C_f^{(\text{add})}$ or $C_f^{(\text{mult})}$. However, to maintain interpretability and align with our primary goal of maximizing discriminative power, we re-rank the final features based on their original positions in the high discriminability feature list. That is, stability is used as a filtering criterion; only features that meet minimum stability requirements are retained, but the final sorting is done according to discriminative strength.

This choice is motivated by the downstream use of these features. These top-ranked features will be mapped directly to font parameters, and not their score values, but the actual feature values will drive font parameter transformations. Therefore, we want these selected features to maximize variation across participants. By keeping them ordered by discriminative power, we increase the chances that different participants will have distinctive feature values for the same font parameters, leading to more diverse and expressive font outputs.

The process of arriving at the final integrated feature set is summarized as follows.

1. Calculate normalized scores: Compute each feature's discriminative score D_f (from the sum of SMD) and stability score T_f (from the chosen stability metric, aggregated as needed across sessions). Normalize these scores to $[0, 1]$.

2. Compute combined scores: For each feature, calculate the additive score $C_f^{(\text{add})}$ and/or multiplicative score $C_f^{(\text{mult})}$ as defined earlier.

3. Rank and select top features: Rank features by the chosen combined score. If both methods are considered, this may produce two ranked lists. Determine the cutoff (e.g., top $N = 6$ features).

4. Re-order by discriminative power: Take the selected features and re-sort them according to their original position in the discriminative score ranking. This ensures maximum between-participant diversity in font rendering.

By following these steps, we ensure that no highly unstable feature slips through, and we avoid overly prioritizing stability at the expense of useful class-separation power. Notably, this procedure reflects the local stability pipeline, meaning that the stability scores T_f used above are derived from within-session consistency analysis. Had we used global stability instead, the T_f values would have been derived from population-wide statistics, as discussed in Section 5.4.3.2.

Figure 5.1 illustrates the entire feature selection process that incorporates the local stability pipeline. The diagram shows how raw data from participant sessions is processed to evaluate discriminability and local stability powers, and then how these metrics are integrated to produce the final feature set. As shown in the figure, each participant's data is partitioned and analyzed for stability locally, then a combined scoring mechanism (such as the additive or multiplicative approach) merges the stability information with global discriminative rankings. The final output is a set of top features that satisfy both criteria.

In conclusion, the integrated feature set represents the intersection of two desirable qualities: high discrimination between participants and high intra-session consistency. These features are ideal candidates for downstream modelling and personalization, such as font generation, where both uniqueness and reliability are essential.

5.6 Results for the Most Discriminative and Globally Stable Set of Features

This section presents the outcomes of integrating discriminability and stability scores to identify the optimal feature sets for both global and local stability scenarios in the context of personalized font generation based on keystroke dynamics. The global stability approach utilizes two distinct methods to combine normalized discriminability and stability scores: an additive weighted sum and a multiplicative product. The additive method assigns weights of 0.7 to discriminability and 0.3 to stability, reflecting a design

choice that prioritizes features that effectively distinguish participants while maintaining a baseline level of consistency across the population.

Remarkably, both combination methods converged on an identical set of the top six features when ranked by their respective combination scores. These features, listed in order of their discriminability scores, are Space (M), Left (M), Right (M), E (M), N (M), and Left+Space (DU). Detailed metrics for these features, including their average values, discriminability scores, normalized discriminability scores, stability scores, normalized stability scores, combination scores, and initial ranks based on combination scores, are provided in Tables 5.2 and 5.3 for the additive and multiplicative methods, respectively.

Table 5.2: Top 6 Features Based on Additive Combination Score (0.7×Disc + 0.3×Stab)

#	Feature	Avg. Val.	Disc. Score	Disc. Norm	Stab. Score	Stab. Norm	Comb. Score	Rank
1	Space (M)	114.62	1255480.18	1.000	0.918	0.551	0.865	1
2	Left (M)	106.97	982734.51	0.767	0.978	0.882	0.801	2
3	Right (M)	90.18	726684.36	0.548	0.954	0.748	0.608	3
4	E (M)	108.67	522039.47	0.373	0.912	0.520	0.417	5
5	N (M)	95.66	446173.01	0.308	0.899	0.446	0.349	6
6	Left+Space (DU)	285.43	445524.23	0.307	0.972	0.850	0.470	4

Table 5.3: Top 6 Features Based on Multiplicative Combination Score (Disc × Stab)

#	Feature	Avg. Val.	Disc. Score	Disc. Norm	Stab. Score	Stab. Norm	Comb. Score	Rank
1	Space (M)	114.62	1255480.18	1.000	0.918	0.551	0.551	2
2	Left (M)	106.97	982734.50	0.767	0.978	0.882	0.676	1
3	Right (M)	90.18	726684.36	0.548	0.954	0.748	0.409	3
4	E (M)	108.67	522039.47	0.373	0.912	0.520	0.194	5
5	N (M)	95.66	446173.01	0.308	0.899	0.446	0.137	6
6	Left+Space (DU)	285.43	445524.23	0.307	0.972	0.850	0.261	4

While the initial rankings are based on combination scores that vary between the two methods (for instance, Space (M) ranked first in the additive method but second in the multiplicative method, with Left (M) taking the top spot), the final feature list was standardized by reordering according to discriminability scores. This reordering aligns with the system's primary objective of maximizing inter-participant differentiation, ensuring that the selected features are those most capable of distinguishing individual typing behaviours. The discriminability scores, which are identical across both tables (e.g., 1255480.18 for Space (M), 982734.51 for Left (M)), dictate the final order: Space (M), Left (M), Right (M), E (M), N (M), Left+Space (DU).

The choice of weights (0.7 and 0.3) in the additive method was deliberate, designed to emphasize discriminability while still accounting for stability. This weighting scheme mirrors the approach used in other components of the system architecture, notably in part B of Figure 5.1, where discriminability and local stability scores were combined in a similar manner.

In contrast, the local stability scenario reveals a markedly different pattern, characterized by greater diversity in the top six features selected for each participant. This variability is illustrated in Figure 5.8, which shows a histogram of the frequency of features appearing in the top six lists across the participant population for both multiplicative and additive combination methods. The histogram highlights that certain features, such as "Left+Left (DD)", "Space+Left (DD)", and "Space+Right (DD)", are frequently selected, each appearing approximately 250 times in the top six lists for both combination methods. Other features, such as "B (M)" or "Right+Right (DD)", appear less frequently, while many, like "C+E (M)" or "M (M)", are rarely or never selected. Notably, the frequency distributions for the multiplicative and additive methods are highly similar, suggesting that both combination strategies identify a consistent set of locally stable and discriminative features across participants.

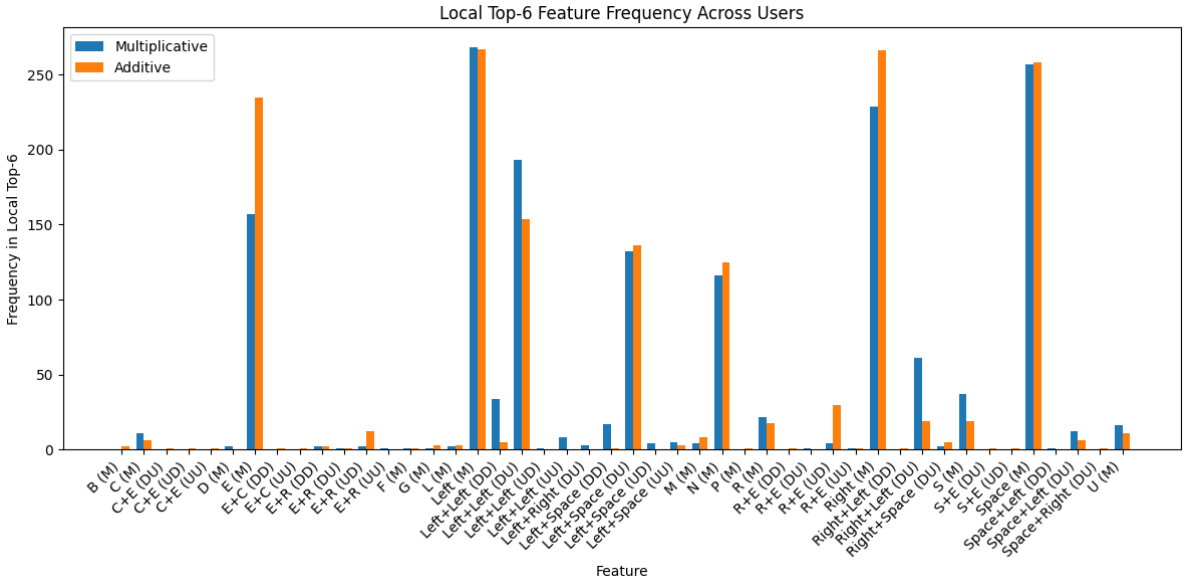

Figure 5.8: Frequency of features appearing in the top six local features across participants for both multiplicative and additive combination methods.

The diversity observed in the local stability scenario underscores the personalized nature of this approach, where feature selection is tailored to individual typing behaviours. These findings have significant implications for the downstream application of these feature sets in font generation. The global stability features provide a standardized set suitable for population-wide models, while the local stability features enable customized representations that capture individual nuances.

The selected feature sets, whether derived from global or local stability approaches, serve as the foundation for translating typing behaviour into personalized font characteristics. In the subsequent sections, we explore how these features are mapped to font parameters, such as weight, slant, and aperture, and ultimately rendered into dynamic typographic representations that encapsulate each participant's distinct typing identity.

This process leverages the discriminatory power and stability of the selected features to ensure that the generated fonts are both visually distinctive and behaviourally grounded.

5.7 User Interaction and Data Capture

The User Interaction and Data Capture component is the system's first point of interaction, where users engage with an interactive interface to input keystroke data, as depicted in Section C of Figure 5.1. Implemented as a web-based application, the interface invites users to type free-form text. After the user has entered sufficient text, they submit their input via the "Submit" button on the interface. Once submitted, the system processes the captured keystroke data and updates the UI with visual feedback. At the bottom of the interface, a "Typing Pattern Graph" is displayed to visualize the user's typing dynamics as shown in Figure 5.9. In this graph, the timestamp of each recorded keystroke is plotted on the x-axis over time. Each keystroke is also plotted at a vertical position corresponding to its character, with the y-axis implicitly encoding key identity based on the order in which each key first appears during the user's typing. Once a key has been used, it is assigned a fixed vertical position that remains consistent throughout the rest of the session. As the same characters are typed again over time, the keystroke markers reappear in those established positions, producing a curved or layered shape in the overall graph that reflects the recurring structure of the user's input.

This representation allows the user to see patterns, such as which letters they linger on (longer bars for keys held down longer) or where there are pauses (gaps in the timeline). For example, clusters of keystrokes with short intervals indicate a rapid typing burst, whereas larger gaps might indicate hesitation or breaks. The Typing Pattern Graph thus serves as an immediate visual summary of the rhythm, speed, and consistency of the user's typing session.

In tandem with the graph, the UI also provides a live preview of the "Visualized Personal Font" panel on the right side of the screen. The generated custom font can be rendered in two ways: on the user's input text or, for comparison, on a standard pangram (the default displayed text by the UI), such as "The quick brown fox jumps over the lazy dog" for a complete alphabet preview. Users can choose between these two text samples using a toggle switch (labelled "Custom" and "Default").

Finally, the user is given the option to download their font via the "Download My Font" button. Clicking this will provide a personalized font file (e.g., an OpenType .otf file) that users can install and use in other applications. In summary, the user interface not only seamlessly captures raw typing data but also visualizes the captured data, both as an analytical graph and as a creative output (the font), thereby helping the user understand and trust how their behaviour is translated into design.

5.7.1 Design Rationale for the User Interface

The user interface design was guided by user-centred design principles, behavioural mapping, and personal expression (as introduced in chapter 3). First, following user-centred design (UCD) principles, the interface is kept simple and focused on the user's primary tasks. The main interactive elements —the text input field, the Submit button, and the Download button —stand out and are clearly labelled (see Figure 5.9). This minimalist layout reduces the cognitive load and makes it immediately obvious how to use the system. Immediate feedback is provided when the user submits a text. This responsiveness keeps the user informed about the system's workings and reinforces a sense of control, as the user can observe the direct results of their actions (typing) without any complex steps in between.

Secondly, the UI incorporates behavioural mapping by translating users' actions into visual outcomes based on their behaviour. The Typing Pattern Graph, for example, helps users make the connection between what they did (for example, a pause or a burst of fast typing) and what is shown on the screen (a gap or cluster in the graph). Similarly, each nuance in the generated font, such as character spacing or stroke thickness, is derived from specific measured features of the user's keystrokes. By designing the interface around this clear stimulus-response pairing (typing behaviour to visual change), the system reinforces the user's mental model of cause and effect, thereby building trust that the personal font reflects their behaviour.

Lastly, the act of creating a personal font turns the user's indiscernible behaviour into an observable, shareable artifact, essentially treating one's typing pattern as a form of self-expression. The UI facilitates this by allowing users to input any text they choose, making the data capture process feel personal and relevant. The resulting font preview immediately shows their own word rendered in their own style, which can be a highly individual experience. By providing the download feature, the interface further empowers users to take ownership of their font, allowing them to integrate a piece of their identity (their "typing signature") into other documents or designs.

5.8 Font Parameters

In order to visually encode typing characteristics as variations in letterforms, we selected seven key typographic parameters to modulate in the generated font. These parameters are weight, slant, width, x-height, contrast, aperture, and taper. Each corresponds to a fundamental aspect of letter shape that can be continuously adjusted while keeping the text fairly legible. In brief, weight controls the stroke thickness (from light to bold); slant adjusts the inclination of letters (from upright to italicized); width expands or compresses the horizontal stretch of glyphs; x-height raises or lowers the relative height of lowercase

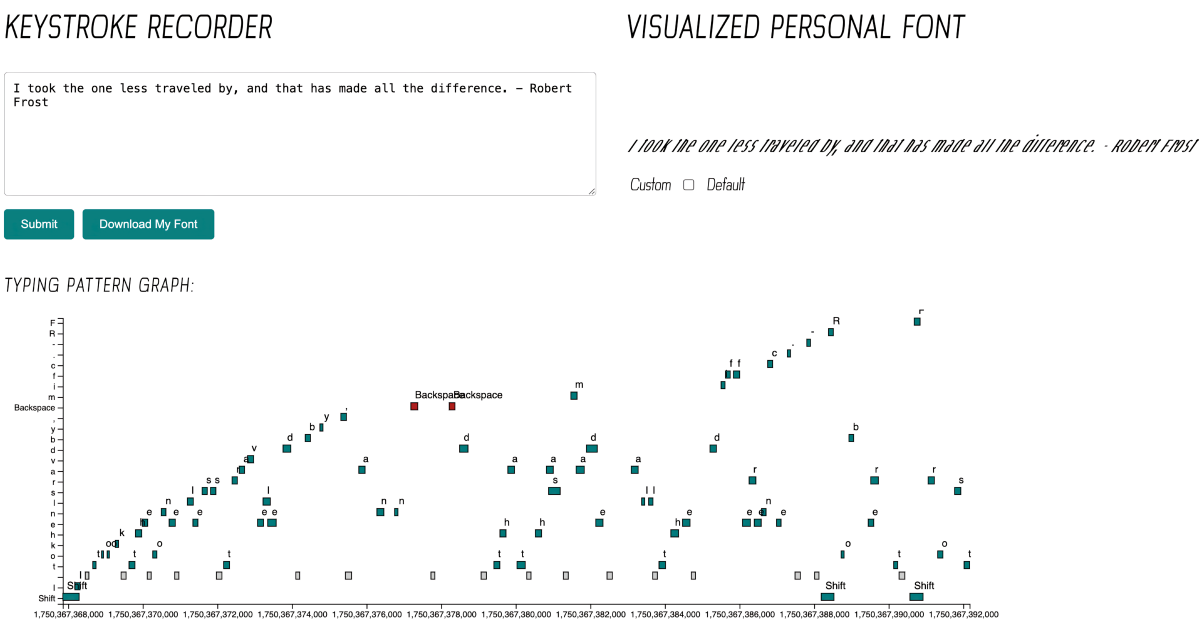

Figure 5.9: User Interface for Typing Pattern Graph: The UI captures keystroke dynamics, displaying the typed text and visualizing the typing pattern graph. This data is processed to generate a personalized font.

letters' bodies; contrast alters the ratio between thick and thin strokes; aperture changes the openness of enclosed letter counters and terminals (e.g., how open or closed a "c" or "e" appears); and taper modifies the gradual narrowing of strokes toward their ends, affecting how pointed or blunt the letter terminals are. These seven font parameters will serve as the visual channels through which personal typing features are expressed in our system's info-typographic representation.

The choice of these particular parameters is grounded in prior empirical research on the perceptual discriminability of font variations, as well as in the practical constraints of our font-generation tool. Recent work by Lang and Nacenta provides an empirical characterization of seven typographical parameters – weight, slant, width, x-height, contrast, serif size, and aperture – regarding how readily readers can perceive changes in each parameter. Their findings indicate that some font attributes are easier for the human visual system to distinguish from others. Based on these results, our design focuses on the former high-observance parameters while incorporating some of the latter for completeness and stylistic breadth. In general, as Figure 5.10 shows, variations in stroke weight and letter slant are the most noticeable, followed by moderate changes in width and x-height, whereas subtle alterations in stroke contrast, serif details and aperture (the openness of letter forms) are comparatively harder to detect [34].

Notably, we omit the specific "serif" parameter and use a taper parameter instead due to the limitations of the font generation platform and their resemblance in defining the letter endpoints. The Metaflop parametric font tool, used for generation (see Section 5.10),

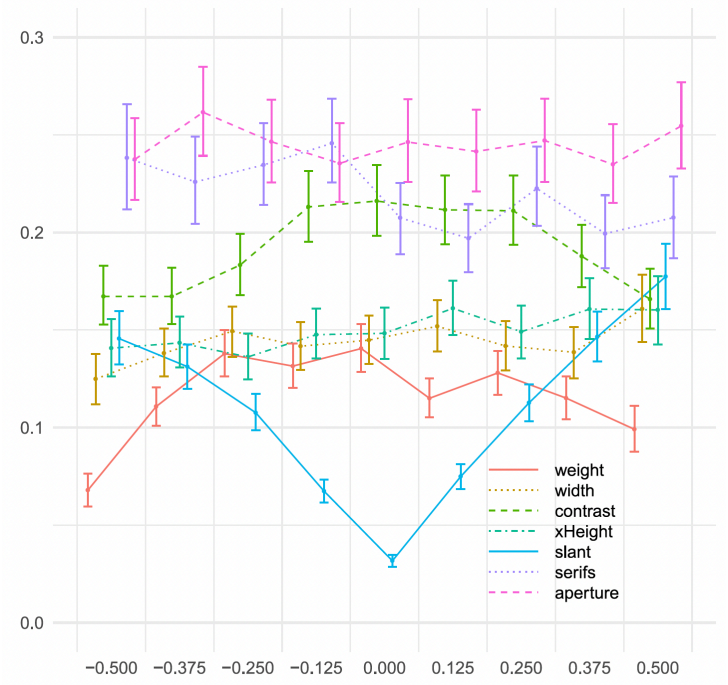

Figure 5.10: Perceptual noise across typographic parameters. Lower values indicate higher perceptual discriminability. As illustrated by Lang and Nacenta [34], serifs and apertures have higher perceptual noise, indicating they are less visually distinguishable compared to parameters such as weight and slant. The image has been used with permission from the author.

does not support dynamically adding or lengthening serifs to letters. Therefore, we substituted taper, which adjusts stroke-end geometry, as an alternative that influences letter terminal appearance in a similar manner. This substitution is also justified by Lang and Nacenta's observations, as serif size and aperture are often grouped as the most difficult to differentiate visually, making it a safe choice in the context of font perception. In other words, both serif and aperture variations tend to be subtle, localized changes (e.g. at letter tips or openings), and the readers in the study showed higher "perceptual noise" (i.e. less sensitivity) in these dimensions. By using a taper control on stroke endings, our font can achieve a comparable effect to serifs (making letters look more or less flared at the ends) while still leveraging the existing axes of the tool. This yields a set of seven controllable font parameters that cover a wide range of visible letterform differences. Each parameter can be manipulated continuously to reflect user data.

5.8.1 Perceptibility and Meaning: The Dual Role of Font Parameters

The primary criterion for selecting these seven parameters—weight, slant, width, x-height, contrast, aperture, and taper—was their perceptual salience, as strongly supported by Lang and Nacenta's empirical ranking of typographic attributes (Figure 5.10). While our system does not explicitly encode semantic intentions (e.g., making text bold

or italic to intentionally signal specific meaning), it is important to recognize that these parameters inherently carry expressive potential, influencing the perceived tone, readability, and personality of the text. Below, we briefly outline the expressive typographic qualities each parameter contributes, drawing on relevant literature to substantiate their suitability for encoding distinctive characteristics.

Weight (stroke thickness) Adjusting a typeface's weight has a pronounced impact on its visual hierarchy and emphasis. Heavier weights naturally draw attention and have long been used to mark importance in typographic hierarchy [10]. At the same time, excessively light or ultra-bold weights can hinder legibility for continuous reading. For example, hefty strokes reduce interior counters, and very thin strokes fade under low contrast conditions. Studies on legibility recommend using mid-range weights for body text to maintain clarity [3]. In summary, the font-weight axis offers a wide expressive range (from delicate to forceful) while remaining perceptible, providing a medium for expressing emphasis or intensity.

Slant (italic angle) The slant of text—upright versus italic—is one of the most salient font transformations and carries established semantic connotations. Italics are often used to introduce motion, contrast, or a distinct voice, distinguishing them from the more neutral and grounded upright forms [10]. Bringhurst emphasizes that italics "express movement and a shift in tone" and are thus inherently expressive even when not semantically intentional. This combination of perceptual salience and typographic function justifies Slant as a valuable axis for modulating the feel and tone of personalized type.

Width (character stretch) Variations in character width—from compressed to extended letterforms—affect the overall rhythm of the text and its expressive tone. Typographic manuals describe condensed styles as applicable in space-limited settings and for evoking efficiency, whereas wider forms tend to appear more relaxed and attention-grabbing [10]. However, overly narrow or wide widths may reduce legibility and reading speed when used in body text. Empirical studies confirm that moderate character widths support fluent reading, while extremes increase visual processing effort [3]. By varying widths within a perceptible but balanced range, we can reflect variation in user typing behaviour without compromising the readability of the output font.

X-height (relative lowercase height) The height of a lowercase "x" (and similar letters) in relation to the overall type height has been well-documented as a factor in text comprehension and size perception. A larger x-height makes lowercase letters taller and generally more legible, especially at smaller sizes or on screen, because the main body of each character is more prominent [3]. Empirical studies have confirmed that increasing x-height

can speed up letter recognition for readers [15] since more of the letterform is within the reader's focal area. In terms of expression, x-height can also influence the perceived tone of a typeface. Fonts with a taller x-height often appear more contemporary, visually open, and approachable, while those with a smaller x-height align with more classical proportions and are associated with a sense of elegance, refined, or formal aesthetics [3].

Contrast (thick–thin stroke contrast) Contrast—the variation between thick and thin strokes in a letterform— affects both perception and typographic tone. High-contrast designs often convey elegance or formality, while low-contrast (monoline) styles suggest simplicity and neutrality [10]. However, extreme contrast can reduce legibility, particularly at small sizes, while moderate contrast helps distinguish letter parts without overwhelming the eye [3]. By modulating contrast, the system can subtly adjust the perceived sharpness or softness of the text in response to user input.

Aperture (openness of counters) The aperture of a letter is the distance between the internal counters of the letter, particularly for letters such as "c," "e," "a," or "s." This parameter subtly affects the tone of openness vs. density in text. There is empirical evidence that open apertures improve recognition. Beier and Oderkerk showed that letters with closed apertures were significantly harder for readers to identify correctly [4]. While aperture alone is a subtle cue, in combination with the other attributes, it contributes meaningfully to the overall typographic personality rendered.

Taper (stroke ends style) The taper parameter controls how much a stroke narrows or flares at its ends, shaping both the visual tone and fine-grained texture of letterforms. Tapered or flared terminals often give typefaces a more refined, humanistic character, referencing calligraphic traditions, even in sans-serif designs. For instance, Hermann Zapf's Optima achieves a soft, elegant tone through subtle tapering, which enhances the aesthetic and functional legibility of the text. In contrast, uncurved, blunt stroke endings produce a more mechanical, impersonal feel [10]. Although taper is a subtle detail, its cumulative visual effect makes a meaningful contribution to readability and expression.

5.9 Mapping Typing Features to Font Parameters

Translating typing features extracted from keystroke data into font generation parameters is a crucial step in personalizing fonts to reflect individual typing behaviour. This process bridges quantifiable typing patterns with the aesthetic attributes of a typeface, ensuring that generated fonts capture the unique characteristics of each participant's keyboard interactions. The primary goal is to map variations in typing style to corresponding changes in font appearance in a consistent and controlled manner. An ineffective mapping

strategy could either obscure subtle behavioural differences, resulting in fonts that lack distinctiveness or exaggerate irregular feature distributions, leading to disproportionate or clustered parameter values. Thus, robust feature-to-parameter mapping is essential to produce personalized fonts that are both visually diverse and faithful to the underlying typing data.

The mapping process involves aligning the most significant typing features with the most perceptually influential font parameters to maximize personalization impact. We mapped the top six features, whether selected globally or locally for their discriminability and stability, to the top six plus one perceptually salient parameters mentioned in the previous section. Due to the close relationship between taper and aperture, the seventh parameter (Taper) is also mapped to the sixth feature value (Left+Space (DU)). This one-to-one correspondence ensures that each top feature influences a specific font parameter, allowing their respective contributions to visual appearance to reflect their discriminability and stability strengths. This systematic alignment enhances the ability to translate typing behaviour into distinct font characteristics while leveraging the most representative features and parameters for the best personalization outcome.

In the case of mapping globally selected features for most discriminability and stability, the mapping would be like this:

Feature	Parameter
Space (M)	Weight
Left (M)	Slant
Right (M)	Width
E (M)	xHeight
N (M)	Contrast
Left+Space (DU)	Aperture
Left+Space (DU)	Taper

Table 5.4: Mapping of globally selected features to font parameters based on discriminability and stability.

Initial experiments with direct linear mapping of raw feature values to font parameters revealed significant limitations. Many typing feature measurements exhibited dense clustering within a narrow range across the participant population. When these clustered values were mapped linearly to font parameters, the resulting parameter values were saturated around the central region of the parameter range, producing fonts that were remarkably similar across participants. This central saturation undermined the goal of font personalization, as subtle differences in typing behaviour were not adequately translated into noticeable variations in font appearance. To address this challenge, a rank-based mapping strategy was developed, transforming feature values into their percentile ranks

within a global population dataset before mapping them to font parameters. This approach ensures a more uniform distribution of parameter values, enhancing the diversity and individuality of the generated fonts. The following subsections detail the ranked approach and compare the distributions of font parameters before and after its application.

5.9.1 The Ranked Mapping Strategy

The ranked mapping strategy was devised to overcome the shortcomings of direct raw-value mapping, particularly the issue of central saturation caused by dense clustering of feature values. In the initial approach, a significant proportion of participants exhibited feature measurements that were closely aligned, resulting in font parameter values converging near the midpoint of the allowed range. This convergence led to a lack of variability in the generated fonts, as many participants received nearly identical parameter settings despite subtle differences in their typing behaviour. The ranked method addresses this by converting each typing feature value into a percentile rank relative to a reference population, which is then used to determine the corresponding font parameter value.

The ranked approach operates in two key steps. First, a globally ranked reference is established for each typing feature, such as Space (M), which measures the mean hold time for the spacebar key. The mean values of Space (M) are collected from all participants in the population dataset, representing the average hold time for each participant's session. These mean values are sorted in ascending order, and each is assigned a unique rank from 1 to N, where N is the total number of participants (e.g., 270). Even if multiple participants share the same mean value, they receive distinct ranks, determined by a consistent tie-breaking rule (e.g., order of appearance), ensuring a fine-grained scale without plateaus. This process effectively spreads the feature values across a uniform rank scale, creating a global reference list where each rank corresponds to a specific position in the sorted list of feature means.

Second, for a new participant, the mean of their Space (M) feature is calculated from their typing session. This value is then located within the global ranked list. If it exactly matches a value in the list, the corresponding rank is assigned. If it falls between two values, say between rank x (with value a) and rank y (with value b), linear interpolation is used to compute the precise rank q:

$$q = x + \left(\frac{\text{user value} - a}{b - a} \right) (y - x)$$

This interpolated rank q is then mapped linearly to the font parameter range, such as 0.1 to 1.0 for weight. For a population of 270 participants, rank 1 corresponds to weight 0.1, rank 270 to 1.0, and intermediate ranks are mapped proportionally.

To ensure fairness, if multiple participants share an identical feature value, they are assigned the same percentile rank, calculated as the average of the ranks they span, hence guaranteeing that identical typing behaviours yield identical font outcomes.

This rank-based mapping normalizes feature values to the population's behavioural spectrum, preventing common behaviours (where many participants' feature values cluster) from collapsing into nearly identical font settings. Instead, these participants receive font parameter values reflecting their slight differences in percentile rank, ensuring diverse font outcomes. While this approach might bring the results for the most repetitive behaviours closer to those of extreme behaviours, the latter occurs less frequently, allowing the strategy to prioritize and increase diversity for the more commonly observed behaviours across the population.

Extreme behaviours are appropriately represented at the parameter range boundaries without exaggeration, with the most extreme participants receiving the maximum stylistic variation and others scaled proportionately. The approach is both deterministic, ensuring that the same typing feature value consistently produces the same font parameter outcome, and interpretable, allowing for a straightforward explanation; for example, a participant's weight parameter might be set based on their feature ranking in the top 15% of the population, clearly linking their typing behaviour to the resulting font design. Although this method may shift the outcomes for the most common behaviours toward those for rare and extreme behaviours, the latter are less frequent, allowing the strategy to effectively increase diversity for the more prevalent behaviours within the population.

5.9.2 Effects of Ranking on Font Parameter Distribution

The transformative impact of the ranked mapping strategy is evident from comparing the distributions of the Space (M) typing feature and its corresponding weight font parameter before and after applying the ranking transformation. The Space (M) feature emerged as the most globally discriminative and stable feature in the integrated feature set, as detailed in Section 5.6. Similarly, the weight parameter, governing stroke thickness, was chosen due to its high perceptual salience, as identified by Lang and Nacenta [34]. Pairing the most discriminative typing feature with a visually impactful font parameter demonstrates the ranking strategy's effectiveness in enhancing font personalization.

Before applying the ranked approach, the distribution of Space (M) feature means across the participant population is multimodal, with peaks at approximately 87.2, 117.8, 133.1, and 163.7 milliseconds, and a median of 115.53 milliseconds (Figure 5.11). The primary peak at 117.8 milliseconds indicates significant clustering, with many participants having similar feature values. This clustering poses a challenge for personalization, as direct linear mapping would result in similar weight settings, limiting font diversity.

This clustering is reflected in the pre-ranked distribution of weight parameter values,

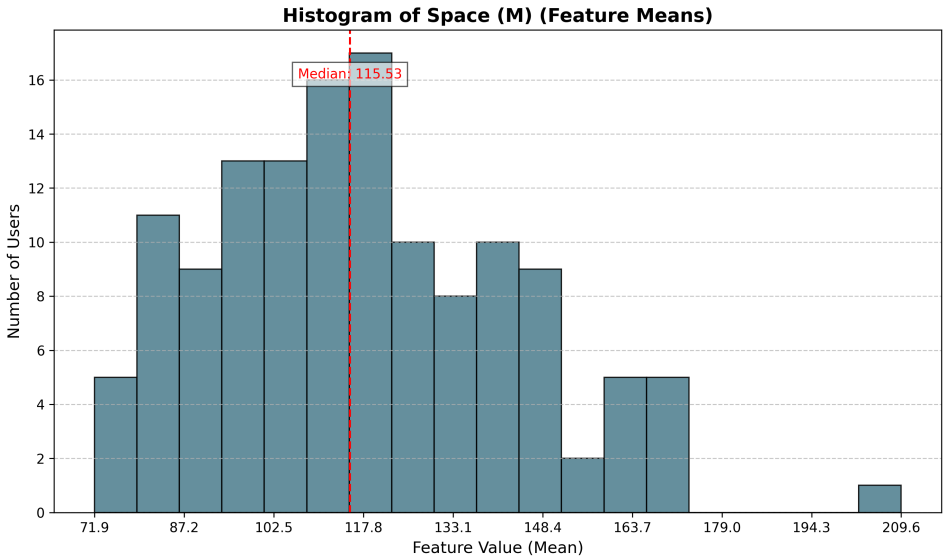

Figure 5.11: Histogram of Space (M) feature means across participants before applying the ranked approach, showing a multimodal distribution with a median of 115.53 milliseconds.

derived from the direct linear mapping of Space (M) (Figure 5.12). The distribution is unimodal, with a peak around 0.620 and a median at 0.60, within a range from 0.210 to 1.440. The concentration around 0.620 mirrors the Space (M) clustering, confirming that direct mapping preserves the feature's distributional characteristics. The median weight value of 0.60 occupies approximately 31.7% of the parameter range (0.210 to 1.440), aligning with the Space (M) median's position (115.53 milliseconds, approximately 31.7% of its range from 71.9 to 209.6 milliseconds). This similarity highlights the direct correspondence between feature and parameter distributions, but the heavy concentration results in limited visual diversity, with many participants receiving similar stroke thicknesses.

After applying the ranked mapping strategy, the weight distribution transforms significantly, as shown in Figure 5.13. The values are now nearly uniformly distributed across the range from 0.1 to 1.0, with a median of 0.55, corresponding to the range's center. Multiple minor peaks, with frequencies varying moderately across bins, indicate that participants are assigned weight values throughout the range, enhancing differentiation. The previously dense cluster around midrange Space (M) values has been dispersed, with slight differences in percentile rank translating into distinct weight settings. Sparse extremes are appropriately represented at the range boundaries, ensuring outlier behaviours receive extreme weight values without dominating the distribution. This flattening effect results from assigning ranks based on value order rather than magnitude, spreading clustered feature values across the rank scale and, consequently, the parameter range.

The shift from a clustered, unimodal distribution to a near-uniform one underscores the efficacy of the ranked mapping strategy. By leveraging Space (M) percentile ranks, the approach ensures weight fully utilizes its range, resulting in diverse stroke thicknesses

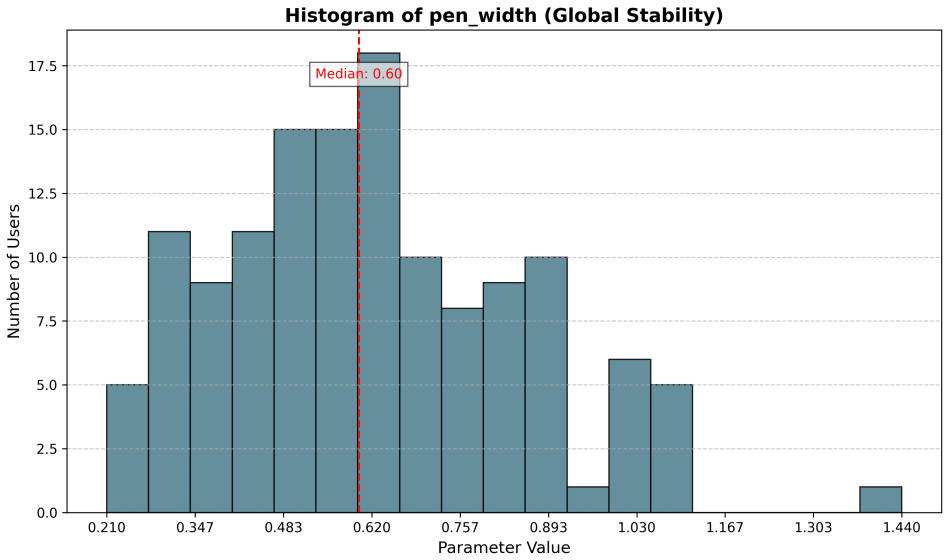

Figure 5.12: Histogram of weight parameter (pen_width) values before applying the ranked approach, showing a unimodal distribution with a median of 0.60.

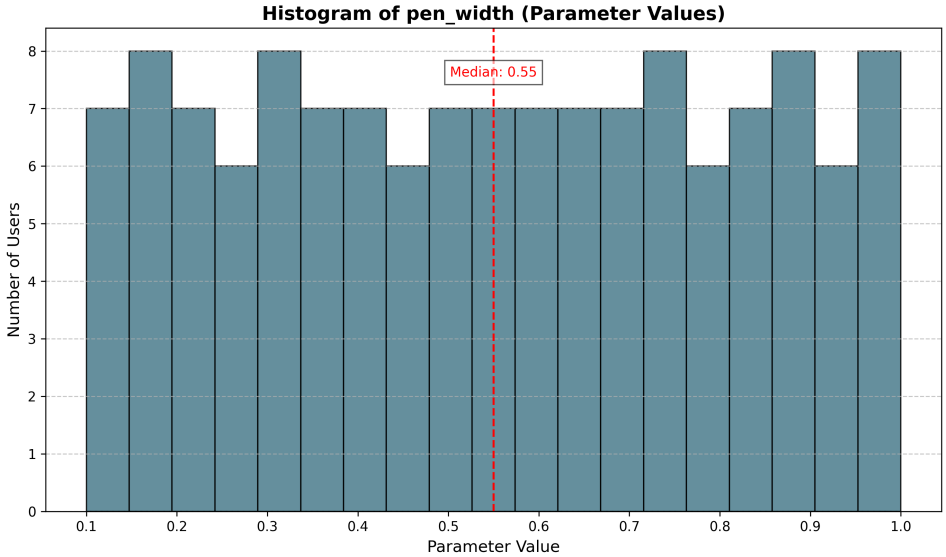

Figure 5.13: Histogram of weight parameter (pen_width) values after applying the ranked approach, showing a near-uniform distribution with a median of 0.55.

across participants. This is significant given weight's perceptual importance, as stroke thickness variations are highly noticeable and contribute to a font's visual identity. The ranked approach maximizes the expressive capacity of the font space, allowing subtle typing differences to manifest as meaningful font variations.

In summary, the ranked mapping strategy effectively addresses direct linear mapping limitations by transforming feature values into population ranks, resulting in a more equitable and diverse font parameter distribution. This ensures personalized fonts accurately reflect unique typing signatures, achieving the goal of creating visually distinct and be-

Table 5.5: Comparison of weight Parameter Distributions Before and After Ranked Approach

Version	Median	Range	Distribution Characteristics
Pre-Ranked	0.60	0.210–1.440	Unimodal, concentrated around 0.620, slightly right-skewed
Post-Ranked	0.55	0.1–1.0	Near-uniform with multiple minor peaks, centered median

haviourally grounded typography. This strategy allows for enhanced personalization by leveraging the most discriminative typing features and perceptually salient font parameters to produce a varied set of font outcomes.

5.10 Font Generation

Once the typing-derived font parameters have been determined (from Section 5.9), the system must generate an actual font file as the next step. Programmatically creating a valid OpenType font (.otf) is a complex task involving outline construction, metrics tables, and font hinting. Rather than implementing a custom font compiler, we outsourced this step to the Metaflop framework. Metaflop is an open-source web application that leverages Donald Knuth's Metafont language to enable parametric font generation [54]. By adjusting a font's design parameters, Metaflop can automatically produce a wide range of typeface variants with minimal manual effort [54]. Crucially, it provides an automated pipeline to export fonts in standard formats: users of Metaflop can download the resulting typeface as an OpenType (.otf) font, suitable for use on any system or application supporting that format [54]. In our system's pipeline (see the architecture diagram, stage D), this Metaflop-based "Font Generator" module takes the mapped font parameter values as input and outputs a custom .otf font file ready for use.

5.10.1 OpenType Font Generation

Generating a fully functional OpenType font through Metaflop involves a non-trivial backend process. Under the hood, the platform links together a chain of tools and scripts to convert a parametric Metafont description into an outline font. For example, Metaflop's server-side workflow employs the Metafont engine and then invokes conversion utilities (such as Linus Romer's mf2outline Python script) to translate the Metafont output into vector outlines and font tables [48]. The mf2outline tool uses FontForge (a font editing library) in the background to auto-trace the shapes and generate an OpenType font file [19]. This approach allowed us to delegate the low-level font construction to proven tools. Every time our system supplies a new set of parameter values to Metaflop,

the backend produces a corresponding .otf font, encapsulating the participant's personal typographic signature in a standard font format. Given the inherent complexity of font generation, Metaflop's open infrastructure (built on FontForge and Metafont) was essential to integrate this step without reinventing the wheel.

5.10.2 Calibration of Font Parameters

Adopting Metaflop introduced some challenges in ensuring the generated fonts were visually appropriate. We discovered that the default parameter values in the chosen Metaflop font template were not visually neutral. In other words, using the template's built-in defaults yielded a typeface with subtle stylistic quirks rather than a plain "average" sans-serif look. If we had naively treated those defaults as the midpoint for our dynamic range, an "average" participant's typing data would produce a font that was unintentionally stylized. To address this, we recalibrated the parameter mappings so that the mid-range value of each font parameter corresponds to a standard, neutral sans-serif appearance. This involved offsetting and rescaling the Metafont's parameters: we adjusted the nominal default of each parameter until the generated glyphs appeared optically balanced and conventional (serving as a baseline). From that calibrated center, deviations in the participant's typing metrics translate to perceptible yet meaningful changes in the font's design. For example, a higher-than-average keystroke stability might nudge a stroke width parameter above its midpoint, making the font slightly bolder, while a lower stability would do the opposite. We carefully set new min/max parameter limits so that these extremes produce noticeable stylistic shifts without distorting the letterforms. Crucially, we calibrated the range for each parameter such that the arithmetic mean of the min and max values equals the neutralized default value, the one that yields a visually standard sans-serif appearance. This ensures that the median participant (with mid-range typing features) generates a font that is perceptually balanced, while deviations from this mid-point (in either direction) result in stylistic variations around a consistent visual baseline. Without this centring, the mapping would risk skewing the output toward one stylistic extreme, making the intended behaviourally grounded differences harder to perceive. After this tuning, the OpenType Font Generation module could reliably turn any set of typing-derived values into a well-formed .otf file, with the mid-point representing a generic sans-serif and the edges yielding visually expressive yet structurally coherent typographic variants.

Therefore, the neutral parameter value for our neutral font is the average value of these min/max values. The parameter ranges and their corresponding neutral values are summarized in Table 5.6. Additionally, a sample of the neutral font generated using these neutral values by the Metaflop tool is shown in Figure 5.14.

Table 5.6: Parameter Ranges and Neutral Values for Font Generation

Parameter	Minimum	Maximum	Neutral (Average)
Weight (Pen width)	0.1	1.0	0.55
Slant	-0.75	0.75	0.0
Width (Unit width)	0.75	2.0	1.375
xHeight (Mean height)	0.5	1.0	0.75
Contrast	1.0	2.0	1.5
Aperture	0.0	0.75	0.375
Taper	0.2	1.0	0.6

Figure 5.14: Neutral font sample generated using the average parameter values by Metaflop.

5.11 Rendering and Presenting the Personal Font

With the custom OpenType font generated, the final step of the system is to render and present this personal font to the user. This component of the pipeline (the rightmost part of stage D in the system diagram) takes the newly created .otf file and integrates it into the user interface. In practice, the system dynamically loads the generated font (for example, by injecting it as a web font in a browser-based UI or installing it temporarily in a desktop application). Then it applies it to a sample text field. The user can immediately preview their personalized typeface in real time. Our interface allows users to type any custom phrase or sentence and see it rendered in their own font, as well as display standard pangrams, such as “The quick brown fox jumps over the lazy dog,” for a full alphabet preview.

This real-time rendering provides instant visual feedback: the abstract keystroke data is transformed into a tangible personalized typography output before the user’s eyes. Notably, this rendering step completes the personalization feedback loop. The system has analyzed the user’s unique keystroke dynamics, mapped those patterns to visual attributes, generated a one-of-a-kind font, and now shows the result back to the user in a familiar textual form. The user interface highlights this final transformation by presenting the Infotypography results in context. For example, as soon as the font is loaded, the UI refreshes the text display area with the user’s font, allowing them to experience their

writing style translated into typography. This immediate application is analogous to how Metaflop's online modulator previews font changes: behind the scenes, a chain of scripts produces an updated font, which is then used to render the on-screen text [54].

In our system, once the font generator produces the .otf file, it is fed back into the front-end, so the user's text is re-rendered in their bespoke font. There may be a brief delay (on the order of a few seconds) during generation and loading, but the outcome is a smooth, live demonstration of the personal font. Beyond on-screen preview, the platform also enables users to download their generated OpenType font for external use. A download link or button is provided in the UI, allowing users to save the .otf file. The user can then install or embed their personalized font in other environments (word processors, graphic design software, websites, etc.), as the font is a standard OpenType file. This capability is important for practical user ownership of the result – it ensures the personalized typeface is not confined to our system. (Metaflop likewise offers both webfont packages and OTF downloads for the fonts created on its platform [54], underscoring the value of portability in parametric font systems.) By obtaining the font file, users can incorporate their "typing fingerprint" font into documents or share it, thus extending the expressive feedback beyond the application itself.

In summary, the rendering and presentation stage is the culmination of the personalized typography pipeline. It provides an immediate, visual affirmation of the user's individuality by showing their text rendered in a font shaped by their own behaviour. This final step reinforces the concept of infotypography by closing the loop: the user's keystrokes influence the font, and the font in turn reflects back to the user in an intuitive visual form. The combination of real-time preview and the option to export the font empowers users to fully appreciate and utilize their personalized typeface, making the system's outcome both experiential and practical.

Chapter 6

System Evaluation & Outcome

This chapter evaluates the outputs of our system by examining the fonts generated for selected test participants. Specifically, we aim to assess whether each font preserves a stable visual identity across multiple partitions of the same participant's data (stability) and whether the fonts generated for different participants are perceptually distinguishable from one another (discriminability).

Given the inherently subjective nature of visual aesthetics, a comprehensive evaluation of stability and discriminability ideally would involve testing actual participants to assess whether they perceive the generated fonts as consistent and distinctive. However, due to practical constraints such as time and participant availability, we conducted an initial qualitative evaluation using a limited subset of test participants. Although quantitative evaluation methods, such as shape similarity metrics or perceptual studies, could offer more formal validation, these were outside the scope of this thesis and are discussed in Section 7.2.5 of Chapter 7. For this chapter, we rely on a qualitative reading of font renderings to explore the extent to which the design system supports consistent and personalized outputs across participants.

We acknowledge that the evaluation ultimately relies on the reader's subjective judgment and will include our observations, informed by repeated engagement with the system and a trained familiarity with the font parameters involved. Wherever possible, we direct the reader's attention to specific traits, such as x-height variation, stroke weight, letter slant, and spacing, which reflect the underlying behavioural mapping. Our intent is not to prescribe a rigid interpretation but to help guide the reader's visual comparison through a consistent and informed lens, highlighting the types of variation that the system is designed to capture.

6.1 Rendered Font Results

As outlined in Chapter 5, Section 5.1, the test dataset comprises 12 participants: 10 were randomly selected from across all available datasets, while two additional samples were collected explicitly from the thesis author (Test Participant #12) and the project supervisor (Test Participant #11). The 12 participants were selected in a manner that ensured their sessions, after feature augmentation, contained more than 11,000 event attribute instances, meeting the minimum threshold established in Chapter 5, Subsection 5.4.1, to be considered in the evaluation. The count of 11,000 refers specifically to the number of event attributes per session, not to the number of unique aggregated features.

For the author and supervisor, sampling was performed manually, with particular attention to the nature of the input. Both were asked to type freely, without following a template, to emulate natural typing behaviour. As with the randomly selected participants, we verified that these manually collected sessions also exceeded the 11,000 event attribute threshold, ensuring their data quality and length were suitable for meaningful comparison and analysis.

The figures below show the rendered font for each of the 12 test participants. In each figure, the results of two participants are presented side by side (in the left and right columns), with their participant IDs labelled at the top. Each column displays the same sample text passage rendered in a personalized font derived from that participant's typing data. This consistent sample text (a quote from Henry IV, used for all participants) rendered from the font files allows for a clear visual comparison of the typeface styles attributed to different participants. Each column displays five font instances: the first instance (labelled "0") corresponds to the whole session, and the following four instances (labelled "1" to "4") correspond to the partitions of that session.

6.2 Comparison

Having presented the font renderings for all test participants, we now compare the results along two qualitative dimensions: stability (the consistency of a given participant's font across different data partitions) and discriminability (the distinctiveness of fonts between different participants). This evaluation serves as a first-level approximation of how well the fonts reflect unique and consistent behavioural signatures.

6.2.1 Stability

Stability, as mentioned earlier, refers to how consistently the system presents a font style for the same participant when derived from different segments of their typing session. This is achieved by treating each partition as a separate session and running it through

Figure 6.1: Rendered font results for Test Participant #1 (left) and Test Participant #2 (right). Each column shows the sample text in the font generated from that participant's keystroke dynamics. The two fonts exhibit noticeably different styles; for example, one appears slightly lighter and more condensed, while the other is bolder with wider letterforms, reflecting the distinct typing profiles of Participant #1 and Participant #2.

Figure 6.2: Rendered font results for Test Participant #3 (left) and Test Participant #4 (right). As before, the same reference sentence is rendered in each participant's custom font. We can observe clear stylistic differences between these two fonts. For instance, Participant #3's font appears relatively upright and evenly spaced, whereas Participant #4's font has a slightly more slanted (italic) appearance with a different weight distribution.

Figure 6.3: Rendered font results for Test Participant #5 (left) and Test Participant #6 (right). The sample text is shown in each participant's generated typeface. Participant #5's font is distinct from Participant #6's font. For example, Participant #5's letters are fairly straight and regular, while Participant #6's letters appear more cursive or inclined. Notably, all of Participant #6's partitions produce visually similar font output, suggesting high internal consistency.

Figure 6.4: Rendered font results for Test Participant #7 (left) and Test Participant #8 (right). Again, the same text is rendered in each personalized font. The two participants' fonts can be readily distinguished: Participant #7's font shows heavier stroke weights and slightly wider spacing, whereas Participant #8's font appears narrower and more compact.

Figure 6.5: Rendered font results for Test Participant #9 (left) and Test Participant #10 (right). The personalized fonts for these two participants display markedly distinct visual traits. Participant #9's font is rendered with thicker, bold strokes and a slightly cursive tilt, whereas Participant #10's font is lighter-weight and more upright.

Figure 6.6: Rendered font results for Test Participant #11 (left) and Test Participant #12 (right). Each participant's distinctive font is applied to the standard test passage. Participant #11's font appears relatively plain and upright, in contrast to Participant #12's font, which is dramatically slanted and bold, giving it a more calligraphic or handwritten character.

the system individually. In the figures above, each column of cards shows five renditions of the same reference text, one generated from the whole session (labelled “0”) and four from equal-length partitions (labelled “1” through “4”).

While visual consistency across partitions seems generally strong for many participants (e.g., Participant #6, whose output maintains a uniform slant and x-height across all five instances in Figure 6.3), this is not universally the case. For example, in the case of Participant #12 (Figure 6.6), noticeable changes in stroke weight, slant, and even letter spacing emerge across partitions. The final instance, in particular (partition 4), exhibits exaggerated angularity and boldness compared to the full-session rendering, suggesting a deviation in at least one of the mapped parameters.

These kinds of fluctuations raise meaningful considerations. One possible factor could be behavioural drift during a session. For instance, some fonts appear more consistent in their early or late partitions, potentially reflecting warm-up effects or end-of-session fatigue. While we refrain from asserting any specific cause, such patterns suggest that typing behaviour is not temporally uniform and that intra-session shifts can propagate into the visual output.

Importantly, it should be acknowledged that the features used for font parameterization were selected based on population-level statistics. While this enables the system to generalize across participants, it introduces a known limitation: population-informed mappings may not be equally robust for every individual. Despite efforts to maximize generalizability, such as selecting globally stable and discriminatory features from a dataset of over 250 participants, the inference mechanism can still falter in some cases. Participant #1 may be one such case, where certain behavioural traits fall outside the norms captured during training, leading to visually divergent partitions such as partition 2, which has a slant that runs in the opposite direction from the rest.

Our view is that these are the expected limitations of population-optimized mapping when deployed on individual-level data. For the purposes of this evaluation, we interpret visual consistency not as exact replication but as the preservation of a recognizable, coherent stylistic identity across partitions. This interpretive lens is informed by repeated engagement with the system and a working familiarity with the font parameters’ behaviour.

6.2.2 Discriminability

Discriminability assesses the degree to which fonts from different participants can be visually distinguished from one another. Based on our observation of the rendered figures, each participant’s font maintains a style that is visually separable from others in the sample set.

For example, the contrast between Participant #2’s broad, bold strokes (Figure 6.1)

and Participant #4's compact, italic font (Figure 6.2) is immediately perceptible. Likewise, Participant #12's highly expressive script-like font (Figure 6.6) diverges noticeably from Participant #11's reserved font structure.

Although fonts such as Participant #4 and Participant #9 may appear loosely similar in slant and shape, close inspection reveals differences in letterform proportions (e.g. aperture and taper) and spacing. To illustrate this distinction more clearly, we include a side-by-side rendering comparison in Figure 6.7. We refrain from making definitive claims about perceptual distinctiveness beyond this participant group; instead, we state that in this limited sample, fonts appear subjectively distinguishable.

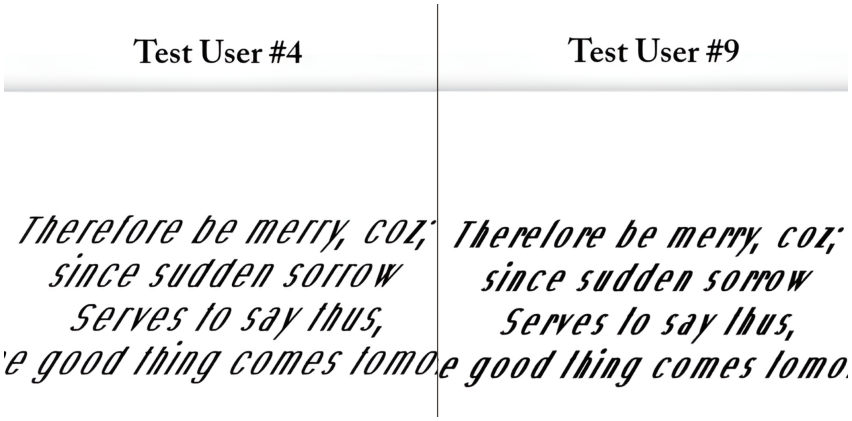

Figure 6.7: Close-up comparison between Participant #9 (left) and Participant #4 (right) fonts. Although both fonts share a similar italic slant, differences in font parameters, such as stroke taper, aperture, and width, become evident.

6.3 Outcome and Interpretation

There are indications that some typographic outcomes align with specific typing behaviours. For instance, Participant #6's font shows an unusually high x-height across partitions, which is consistent with the long key-hold duration recorded for key 'e,' the feature used to drive the mean height parameter in the system (see Chapter 5, Table 5.4). Similarly, the broad spacing in Participant #2's font corresponds to a longer time spent pressing the monograph keys on the right side of the keyboard. These examples illustrate how certain behavioural traits may be meaningfully expressed in the font's visual structure.

The results suggest that the system preserves internal visual consistency and generates relatively distinct fonts across individuals. These findings support the principles outlined in Chapter 3, particularly the use of participant-centred and data-driven design to map behavioural data to visual expressions through generative font modification. As a result, the system's performance across all twelve test participants can be viewed as proof of concept for personalization based on behavioural input.

Chapter 7

Discussion and Future Work

Our system successfully generated personalized fonts for each test participant, visually reflecting individual typing patterns. In qualitative evaluations, the fonts remained largely consistent across multiple data partitions for the same participant (demonstrating stability) and exhibited noticeable stylistic differences between different participants (demonstrating discriminability). For example, some participants' fonts appeared bolder or more slanted than others', aligning with the distinctive aspects of their typing behaviour. At the same time, the degree of visual contrast varied as specific fonts were more similar than expected, indicating that the font-generation approach arguably constrained the range of achievable styles.

These results suggest that keystroke dynamics can encode personal traits into a visual form, providing a new mode of self-expression in typed text. The fact that a participant's font maintains a cohesive style across their own typing samples implies that our selected typing features capture the stable, intrinsic aspects of that individual's behaviour. Meanwhile, differences between participants' fonts confirm that typing patterns carry identifying information, much like a biometric signature, even though all participants may be typing the same content. This bridges the gap between impersonal digital text and the individuality of text visuals, showing that electronic communication can be personalized. Meanwhile, subtle overlaps in some fonts highlight the current limitations of the system and suggest future opportunities to expand font parameter space and evaluate fonts more rigorously.

7.1 Connections to Prior Work

Our approach intersects with multiple areas of existing research. We interpret the results of our work in terms of three contexts: keystroke dynamics as a biometric modality, parametric font generation tools, and handwriting-to-font technologies. This discussion highlights how our contributions draw from and differ from these lines of work.

7.1.1 Keystroke Dynamics in Biometrics

Keystroke dynamics have long been studied as a form of behavioural biometric for user identification and authentication. Prior works have shown that individuals can be recognized by statistical patterns in their typing, such as key press durations and inter-key intervals [29]. For example, Kasprowski et al. demonstrate in Sensors 2022 that machine learning models can achieve high accuracy in identifying participants based on the timing features of keystrokes. Such biometric studies typically treat typing patterns as a secret or identifying signal, focusing on security applications like continuous authentication or fraud detection. By contrast, our work repurposes keystroke dynamics for creativity and personalization rather than security. Instead of verifying identity in the background, we visibly manifest a participant's typing pattern in the form of a font. This shifts the perspective from using keystroke timing as a hidden authenticator (with strict requirements on consistency and resistance to mimicry) to using it as a source of individualized design. Nevertheless, we benefited from the biometric literature's insights: features that are distinctive and consistent enough to identify a participant are naturally good candidates for driving unique and stable font generation. Our system's premise builds on the core finding of keystroke biometrics—that typing behaviour contains a recognizable signature—and channels it into a novel medium for personal expression.

7.1.2 Parametric Font Generation Tools

Our project also relates to existing tools and research into parametric font generation. Traditional font design is labour-intensive, but systems like Metafont (Knuth) and modern successors such as Metaflop and Prototypo allow fonts to be defined by parameters that can be adjusted to produce variant typefaces. Metaflop in particular provides an open-source parametric font platform [54], which we leveraged through the 'Bespoke' template. Prior to our work, parametric font tools had been used for rapid type design and personalization, but typically with direct participant input (such as sliders or interactive adjustments) rather than linking to an external dataset. Prototypo [22] is another system that enables participants to interactively tweak sliders (for x-height, weight, width, etc.) to create bespoke fonts without manual drawing. These platforms demonstrate the range of visual diversity that can be achieved by tuning a set of underlying font parameters. We extend this concept by driving the parameters with behavioural data: instead of a designer or end-user choosing the values, the values are determined by the user's typing measurements.

Recent research has begun to explore advanced methods for navigating font parameter spaces. For instance, AdaptiFont by Kadner et al. uses a generative font model and Bayesian optimization to tailor fonts for optimal reading speed, effectively personalizing a font for each reader's performance [28]. Their system adjusts parameters in multiple

dimensions (weight, spacing, etc.) to maximize a measurable outcome (reading speed), which parallels our goal of adjusting parameters to reflect a measurable personal signature. Another example is the work of Tatsukawa et al., who developed FontCLIP, a model connecting vision-language embedding (CLIP) with typography to enable semantic font modifications [53]. FontCLIP can adjust a font's style in response to descriptive prompts (e.g., "playful" or "formal"), showcasing a different form of parametric control driven by high-level semantics rather than participant-specific data. Compared to these, our approach is unique in using biometric-driven inputs: we map low-level timing metrics to visual attributes, introducing implicit personalization where the participant's behaviour guides the design without explicit semantic intent or iterative feedback. This illustrates a novel crossover between human-computer interaction data and parametric typography.

7.1.3 Handwriting-to-Font Conversion

An alternative route to personalized typography is to convert a person's handwriting or hand-drawn letterforms into a digital font. Commercial tools (e.g., Calligraphr) enable participants to scan predefined letter templates to generate a font that resembles their handwriting. In research, this theme evolved rapidly with the advancement of machine learning. Mitrevski et al. introduce InkSight, a system that can "derender" offline handwriting (scanned ink on paper) into an online format (vector strokes), essentially teaching vision-language models to read a static handwritten page and reproduce it as a vectorized digital ink sequence [38]. This enables the creation of a font or digital ink that captures the style of one's handwriting, bridging the gap between pen-and-paper note-taking and digital text. Likewise, other projects have employed generative networks to learn a manifold of handwriting styles and produce novel characters in a participant's handwriting style.

Our work contrasts with these handwriting-to-font efforts in that we do not use samples of a person's writing or drawing. The participant's physical handwriting, including letter shapes and stroke order, is not taken into account. Instead, we derive pseudo-handwriting (the font) from an entirely different signal (typing rhythm). In a sense, handwriting-to-font systems preserve the visual identity of a participant's script, whereas our system aims to convey a behavioural identity through a new visual form. However, both approaches share an ultimate goal: allowing individuals to have a unique font that represents them. An interesting connection is that both keystroke dynamics and handwriting capture personal style, one through timing and motion, the other through form. Future systems might even combine these avenues, for example, using typing behaviour to modulate or select from a palette of handwriting-like fonts.

Handwriting-to-font conversion preserves direct visual identity but lacks dynamism and adaptability. Our approach captures behavioural identity through typing patterns,

allowing fonts to reflect situational states and changes over time, even though they may not reproduce literal handwriting forms. We also note that our findings complement vision-based approaches. While InkSight focuses on accurately preserving the appearance of handwriting, our focus is on generating distinguishable personal styles from abstract data.

7.2 Implementation and Data Limitations

While our results are promising, it is essential to acknowledge the limitations arising from our implementation choices and dataset. Several factors inherent to our system and methodology constrain the expressiveness or consistency of the generated fonts. Addressing these issues will be crucial to improving the system's robustness and real-world viability.

7.2.1 Font Generation Tool

One fundamental limitation originated from the font generation back-end. We selected Metaflop as the font-generation engine due to its open-source accessibility and callable library interface. However, the most complete and stable Metaflop font template available for our needs —MF Bespoke— came with a stylistic restriction: it did not support serif terminals (the small decorative strokes at character ends). In fact, Bespoke is geared towards constructed sans-serif fonts [54], meaning none of the generated letterforms could have serifs or traditional flared stroke endings.

Serifs are a major differentiator in type design, and being unable to toggle between serif and sans-serif styles meant that an entire dimension of typographic expression was off-limits. Consequently, the fonts generated for different participants tended to share a similar minimalist aesthetic, varying only in more subtle geometric ways. For example, one participant's font could not capture the "bookish" feel of a serif typeface, while another's was a clean sans-serif; instead, all participants' fonts were variations on a sans-serif theme. This inherently narrowed the range of styles our system could produce. All the personalized fonts in our study fell within the sans-serif category, without the option to incorporate serif characteristics.

Looking forward, addressing this limitation would involve expanding the font-generation toolkit to include serif-supported or more stylistically versatile templates. One approach would be to extend Metaflop's library (or a similar parametric font system) with a base font that includes serifs or other distinctive features. If such a base had been available or developed, our system could have mapped certain participant traits to the presence of serifs (or other stylistic switches), immediately broadening the visual differentiation between these fonts. In the absence of an existing solution, future work may explore

creating a custom Metafont template or utilizing alternative font-generation frameworks that incorporate a richer variety of letterform styles. This would allow personalized fonts to span a more expansive design space (e.g., serif vs. sans-serif, high contrast vs. low contrast), thereby better capturing each participant's identity in a visually distinctive manner.

7.2.2 Available Parameters

Another factor influencing the expressiveness of the generated fonts was the limited number of adjustable parameters actively incorporated into our system. The Bespoke font base, provided by Metaflop, offers 14 tunable parameters that control different aspects of the typeface's geometry [54]. These parameters cover categories such as dimensions (e.g., stroke thickness, character width), proportion (e.g., x-height, ascender/descender length), and shape (e.g., slant, curvature). However, while Bespoke exposes this parameter set, our system deliberately utilized only a subset of these parameters.

The primary reason for this selection was to focus on parameters with high 'textual salience', i.e. features where changes are visually significant and easily distinguishable by the human eye. For example, stroke weight and slant are readily noticeable, whereas parameters like overshoot or internal curvature adjustments often produce more subtle visual differences. Consequently, out of the 14 available parameters, only seven were mapped to keystroke-derived features in our implementation, prioritizing those most likely to create noticeable variation in personalized fonts.

This design choice, while grounded in maximizing perceptibility, inherently constrained the system's degree of freedom. With only seven parameters influencing the output, the space for possible font variations was bound, limiting how distinct each participant's font could appear. Since all seven parameters were derived from typing behaviour, any relevant differences between participants that could not be meaningfully mapped onto these selected font characteristics remained unexpressed.

It is also important to note that, beyond our selective mapping, the Bespoke font base itself imposes structural constraints. Even with all 14 parameters engaged, the design space remains narrower than the full diversity observed across existing typefaces or handwriting styles. Specific stylistic dimensions, such as serif presence, decorative elements, or extreme contrast, are absent from the Bespoke template, limiting the scope for personalization regardless of parameter count.

Looking forward, increasing the number of adjustable parameters integrated into the system could provide additional freedom to capture typing differences, but parameter expansion alone is insufficient. Any additional parameters must also meet the criteria for perceptual impact; otherwise, they would contribute minimal visible variation, even if they are numerically different. Future work could explore leveraging other Metafont

bases or parametric font systems that offer a broader diversity of salient parameters. This would allow the system to encode a wider range of behavioural traits into visually meaningful font attributes, thereby enhancing the expressiveness of personalized outputs.

7.2.3 Parameter Range

Even when a particular font parameter is available and correlates with a participant-specific trait, the numerical range of that parameter can impose another limit. Each adjustable font parameter operates within a specific range, beyond which the font's design would break or become incomprehensible. For example, the stroke weight parameter might only range from a thin minimum (producing a lighter typeface) up to a thick maximum (beyond which strokes would merge or letters would lose legibility). Similarly, parameters like X-height or slant have built-in bounds to keep the font within plausible shapes. This means there is a cap on how much a font can be pushed or pulled in any given dimension. Once a participant's derived value for a parameter reaches one of these extremes, it cannot exceed it further, and the parameter has effectively saturated.

Within this finite range, the system must distribute all participants' fonts along the available span. To reduce the risk of crowding, we applied ranking-based normalization to spread participants' parameter values across the range, regardless of how clustered the raw values initially were. This helped maximize the separation between fonts in the current population dataset, which forms the reference for future participants. Once this population parameter space is established, every new user or participant's feature value is mapped to the closest available parameter value already present in the population set. However, as the number of participants increases, this solution becomes less effective: with more parameter points packed into the same fixed range, the spacing between assigned values naturally shrinks, and the probability of two participants receiving nearly identical parameter values rises. Even when two participants have distinct typing patterns, their mapped fonts may look visually alike, simply because of range saturation and finite resolution. While our study used 214 population participants and 12 test participants, this limitation becomes far more pronounced at larger scales, for example, if the system were deployed for thousands or millions of participants.

Addressing this limitation by extending parameter ranges is theoretically possible, but it is practically constrained. While minor adjustments beyond tested limits may slightly stretch differentiation, excessive range extension risks producing broken or illegible fonts, thereby defeating the purpose. The inherent boundedness of font parameters remains a structural limitation that cannot be entirely eliminated by scaling alone.

Alternative strategies, such as introducing additional tiers of variation after primary parameters saturate, are also problematic in our context. The parameter mapping occurs during population dataset processing and is fixed relative to those participants. Adding

secondary parameters post-hoc would not resolve collisions among new users unless the entire population is recalculated. Furthermore, key parameters, such as stroke weight, are deliberately chosen for their high perceptual salience; replacing or supplementing them with lower-impact parameters dilutes the distinctiveness of the results. Subtle parameter changes may not be noticeable enough to distinguish fonts meaningfully, especially for untrained observers.

Therefore, while techniques like slight range expansion or secondary parameters offer marginal benefits, they cannot fully overcome the limitations imposed by fixed parameter ranges, particularly as the system scales. Expanding the diversity and number of high-impact, perceptually meaningful parameters (as discussed previously) remains the more viable long-term solution to preserving font distinctiveness across a growing participant population.

7.2.4 Population Dataset and Sampling

The characteristics of the population dataset used for personalization directly affect how input features are mapped to font parameters. Because the system uses the distribution of feature values from this population to determine how test participants' or new users' typing metrics translate into font outputs, any biases or limitations in the dataset can propagate into the personalization process. Several factors, including the dataset size, the quality of the recorded typing sessions, and the nature of the typing tasks, can impact the final font outcomes.

7.2.4.1 Dataset Size

The combined population dataset, comprising 214 participants drawn from the UB and Timisoara keystroke datasets, was adequate for the scope of our study. It offered a reasonably diverse sample of typing behaviours and provided a proper distribution of feature values for initializing the font parameter mapping. As described in Chapter 5, the system first analyzes this population dataset to compute values for the selected top features. It distributes those values across the available range of each corresponding font parameter. For every future user or test participant, the system then maps their feature values to the nearest parameter values already populated by the reference set.

While a larger population dataset could offer denser coverage and potentially more precise mappings for incoming users, fixed parameter ranges would still present limitations. As discussed in Section 7.2.3, the more test participants the system accommodates over time, the higher is the likelihood that multiple cases will be mapped to remarkably similar font parameter values, particularly when their typing features fall near each other in the already saturated space. In fact, even with 214 participants, we observed early signs of this saturation, where some personalized fonts became only subtly distinguish-

able. Therefore, while the dataset size positively impacts the granularity of the mapping, it does not address the broader issue of finite resolution within a bound parameter space.

7.2.4.2 Quality of the Typing Sessions

The quality of individual typing sessions determines the reliability of extracted behavioural features. Since the datasets used in this study were obtained from existing sources rather than explicitly collected for our system, we had limited control over the data collection conditions. Some sessions may have been affected by participant fatigue, distraction, hardware variability, or inconsistent typing contexts, all of which can introduce noise or atypical patterns into the recorded keystroke data. Low-quality sessions reduce the reliability of derived feature values, particularly when those values are used to establish population-wide distributions for mapping font parameters. We applied outlier removal techniques to mitigate the most extreme deviations in keystroke timing in our implementation.

While such limitations were acceptable within the scope of this work, future efforts that involve primary data collection should incorporate stricter controls to ensure session consistency and ecological validity. This includes designing instructions to elicit natural typing behaviour, filtering for sustained attention, and standardizing hardware or environments where feasible. Improving session quality would help ensure that extracted features reflect genuine and characteristic aspects of participant behaviour. This would enable the generated fonts to more accurately capture each participant's unique typing style.

7.2.4.3 Typing Tasks: Fixed vs Dynamic Text

Another influential factor is the nature of the typing task used to gather population data, specifically, whether participants typed a fixed, predetermined passage or engaged in dynamic, free-text entry. In our study, we employed dynamic text tasks to capture natural, real-world typing patterns better. Free-text input allows participants to type at their own pace, select their own wording, and exhibit authentic behaviour such as pausing, backspacing, or hesitating. These elements contribute to a richer behavioural signal, which aligns with the objective of generating fonts that reflect a participant's unique typing signature. By contrast, fixed-text tasks often elicit more mechanical or constrained behaviour, as participants focus on accurately copying content rather than expressing themselves naturally.

However, using dynamic text introduces variability across participants. Since each participant generates different content under potentially different contexts or emotional states, the resulting feature distributions are more challenging to interpret in a strictly comparative sense. Fixed-text input, however, allows all participants to type the same

material, offering a more controlled basis for evaluating differences in keystroke dynamics. This control improves the interpretability of measured features and supports more consistent mappings between behaviour and font parameters. The trade-off, then, is between ecological realism and experimental control. In our case, we prioritized capturing real behavioural traits, but future extensions could benefit from controlled comparisons between fixed and dynamic input modes to evaluate how the nature of the task influences both the feature quality and the resulting fonts.

7.2.4.4 Language Composition

Our dataset included a Romanian free-text corpus from the Politehnica University of Timișoara to enlarge the population and capture unconstrained typing. Because character and digraph distributions vary by language, several English digraph combinations can be uncommon or absent in Romanian, which means that their sample size effect in the pooled dataset will be reduced. As a result, estimates of stability and discriminability for digraph-based measures (e.g., flight-time features) may have been less reliable than for monograph timings. This consideration may partly explain why four monograph features were ranked higher in the top six compared to digraphs. Future work could address this by stratifying analyses by language or balancing samples.

7.2.4.5 Participants vs Actual Users

A notable limitation arises from our use of participant data instead of actual user data. The distinction is significant because participants in our study might not have been fully aware or intentional in influencing the final visual appearance of their font, potentially leading to typing patterns that differ subtly from those of actual users who engage consciously and intentionally with the system. Actual users, aware that their typing directly shapes their personalized font, might deliberately alter their typing behaviour to produce desired visual outcomes, something participants in our study did not necessarily consider.

This limitation suggests that our results, while robust in identifying stable and distinguishing features across participants, might not fully capture the interactive and intentional nature of actual users' engagement with the system. Therefore, fonts generated based on participant data may not entirely represent the dynamics of real-world usage. Future research using actual user data could further refine the system by accounting explicitly for intentional and adaptive typing behaviour.

7.2.4.6 Operating System's Timing Precision

An external factor identified during our analysis was the interaction between the population dataset and the operating system's event sampling frequency used during data

capture. In particular, we found that the inter-keystroke intervals recorded for each participant exhibited a quantification effect imposed by the system clock.

The histogram of timestamp differences, shown in Figure 7.1, illustrates the time intervals between consecutive keystroke events across all participants in our accumulated dataset, which combines sessions from both the UB and Timisoara datasets. These participants represent a wide variety of keyboards and system environments. All sessions involved dynamic text entry tasks (i.e., free typing, not copying a fixed passage). For this analysis, we filtered out participants with inconsistent or abnormal keyboard configurations to minimize hardware-induced variance and more accurately isolate system-level effects. Additionally, timestamp differences greater than 300 ms were excluded due to their low frequency and high likelihood of being outliers.

The histogram revealed several distinct peaks, which correspond to common inter-keystroke intervals during natural typing. Notably, the distribution exhibits tight clusters of peaks interspersed with gaps, where certain intervals never occur. This irregular structure is consistent with the quantization effect introduced by operating system-level sampling mechanisms. Modern operating systems and keyboards typically use fixed polling rates (e.g., 125 Hz, equivalent to 8 ms intervals) or limited timestamp precision, which can force timestamp values to align with discrete intervals. This quantization leads to observed clustering and empty bins in the histogram, reflecting how system-level event logging may shape recorded typing behaviour.

This kind of timing quantization is a known artifact in keystroke data collection. Other researchers have observed and cautioned about similar clustered interval patterns when using typical high-level event timers. For instance, an analysis of free-text keystroke dynamics collected via a web browser noted that the inter-key intervals often fell into multiples of approximately 30 ms, with the data forming spikes separated by $\sim$30 ms due to the underlying timer resolution [13].

In our case, the operating system's event sampling frequency (likely on the order of a few milliseconds) creates a temporal granularity that limits how precisely we can measure typing rhythm. If two participants have slightly different typing speeds or hesitations (say one typically waits 105 ms between keystrokes and another 110 ms), but the OS clock quantizes both to 0.11 s. From the system's perspective, their behaviour appears identical in that regard. The more participants we have, the more this effect becomes apparent. With a large population, the natural timing differences between individuals will likely coincide with the same measured values due to the quantization effect. Thus, the fidelity of captured behaviour is reduced. Clusters of identical interval readings in the dataset correspond to clusters of similar parameter inputs for the font generator, which in turn can result in undesirably similar fonts.

The implications for our system's results are that some of the subtle differences between participants' typing patterns were probably lost or blurred. This could partially

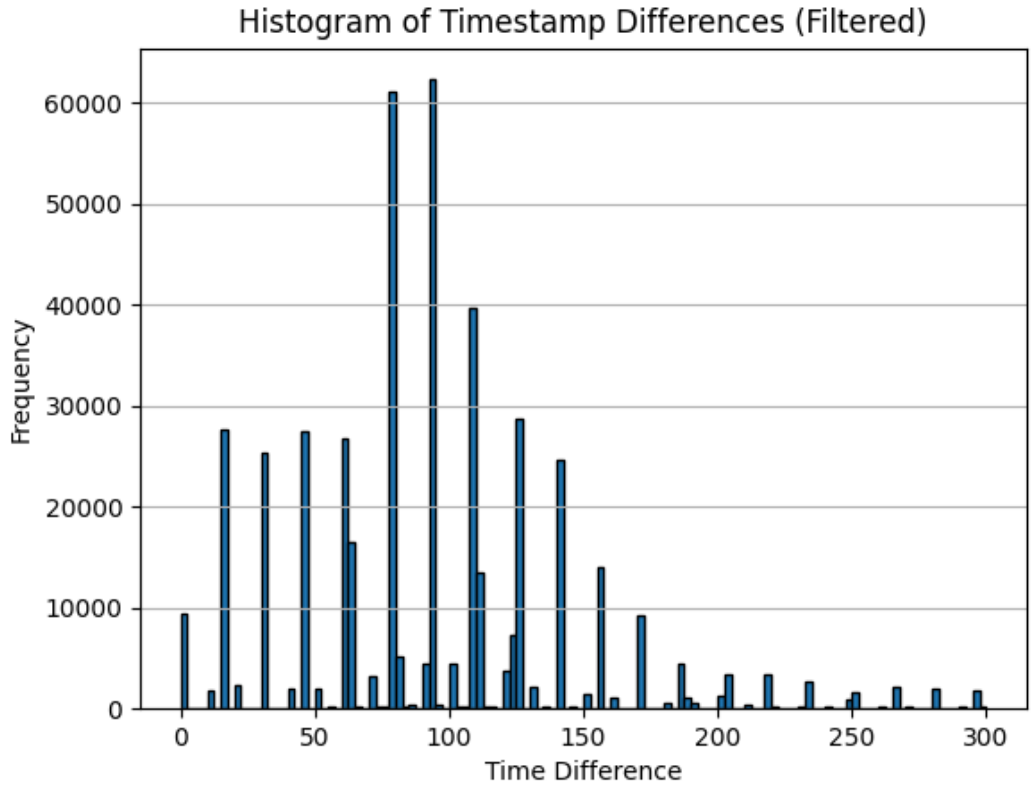

Figure 7.1: Histogram of timestamp differences between consecutive keystroke events, aggregated across all participants and sessions in the accumulated dataset (UB and Timisoara combined). The Y-axis shows the total count of each time interval across the dataset. The sharp peaks and intervening gaps reflect the influence of OS-level sampling precision and polling rates on how inter-key timings are recorded.

explain why the personalized fonts of certain participants did not differ as much as one might expect. If their typing profiles were rendered too similar as a result of coarse time sampling, the font generation stage had little basis for differentiating them.

To improve on this aspect, future implementations could aim for a higher timing resolution when recording keystrokes. This could involve using low-level hooks or high-resolution timers provided by the operating system (for example, using performance counters or raw input APIs that record time in microseconds) instead of relying on default event timestamps. By capturing more fine-grained timing data, the system would be able to detect actual differences in typing rhythms that are currently being masked.

7.2.5 Quantitative Evaluation

To confirm that personalized fonts truly capture meaningful behavioural variation, more objective evaluation strategies can be explored. One direction is to employ glyph-level similarity metrics and morphological analysis tools that quantify geometric differences between letterforms. For instance, contour-based shape descriptors or outline clustering

methods could be used to measure the distance between two fonts in the design space. Such metrics would help determine whether the visual distinctions between different participants' fonts correspond to measurable structural variations in their glyphs, rather than arbitrary stylistic variations.

Another valuable approach involves perceptual participant testing. For example, participants could be presented with a triad of font samples and asked to identify which two fonts appear most similar to each other. This triadic comparison method, commonly used in perceptual similarity studies, would reveal whether fonts generated by the same participant tend to be grouped together by observers. Consistently grouping a person's font variants closer to each other (and away from others' fonts) would indicate that personal fonts have a distinct visual identity aligned with that individual's typing signature.

Additionally, computer vision-based models can automate the assessment of font distinctiveness. For instance, a neural network trained to embed fonts into a perceptual feature space (analogous to style recognition or font classification networks) could estimate similarity scores between fonts. Alternatively, by leveraging a font "style transfer" model or a deep metric learning approach, one can obtain an embedding where the distance correlates with the human-perceived difference. Using such vision models on a large font dataset would enable scalable validation of our fonts' uniqueness.

7.3 Speculation on Real-World Use

Beyond the immediate scope of our experiments, several open questions remain regarding real-world deployment, user adoption, and the broader implications of personalized fonts. Here, we speculate on these aspects and highlight the challenges on the path from a research prototype to practical, widespread technology.

One set of questions concerns real-world deployment. How might a keystroke-dynamics-based font system be integrated into everyday digital life? One possibility is to build it into messaging platforms or operating systems, where, as you type, your personal font is automatically applied to your text, allowing recipients to see it. However, this raises technical hurdles: performance and latency would need to be addressed (generating or retrieving the font on the fly without noticeable delay), and standards for embedding or sharing the font across devices would be required. There are also compatibility concerns. For example, if a recipient's device cannot access or trust the custom font, communication could fall back to defaults, losing the personal touch. Thus, achieving seamless use of personal fonts in communication would require coordination between font technology and communication protocols, perhaps via embedding fonts in emails or utilizing web font delivery for chat applications. Ensuring security and privacy during deployment is another factor (fonts derived from biometric data should not inadvertently leak sensitive information). These considerations suggest that, although the concept is attractive, im-

plementing it ubiquitously would be nontrivial.

Another open question is user adoption and experience. If technology were available, would people want to use it, and in what contexts? On the one hand, personalized fonts might be welcomed in personal and creative communications, much like custom avatars, emojis, or handwriting-like digital stickers that people use today. It could allow users to brand their messages with their identity in a subtle way. On the other hand, in professional or formal settings, a highly distinctive font might be seen as inappropriate or distracting. There is a social dimension to consider: if only one person in a group email uses a personal font, does that come across as egotistical or playful? If everyone uses personal fonts, does it enhance expression, or does it create visual confusion? User studies on acceptability and preferences would be valuable in determining the answer. Perhaps users would want the ability to turn the feature on or off depending on context, or to have multiple persona-based fonts (e.g., a "casual font" for friends and a restrained variant for work). Additionally, there's the question of whether people identify with the font produced from their typing. Do they feel it truly represents them, or would they desire to tweak it (which then blurs the line between automatic generation and manual design)? Since our current system doesn't involve user feedback in the loop (the font is generated without user input beyond typing), it remains to be seen if end-users are satisfied with the outcome or if they would want some control (for example, an option to "dial up" or "dial down" certain stylistic aspects of their font).

From a research standpoint, an open technical challenge is to enhance the mapping from behaviour to a font beyond the straightforward approach we took. We used a relatively simple and one-to-one mapping of features to parameters. Is this the best way to capture the nuances of someone's typing? It's possible that a more complex or multidimensional mapping strategy could uncover richer personalization. For instance, perhaps certain combinations of typing features (a particular pattern of speeds and pauses) correspond to a high-level style trait in writing, which could be reflected in a coordinated change in multiple font parameters. Machine learning models can be trained to discover such relationships, rather than relying on manually engineered mappings. However, doing so would require much more data to avoid simply fitting noise. Another intriguing direction is to incorporate temporal patterns or sequences (n-gram patterns of keystrokes) rather than just aggregate statistics. Our approach reduces a user's typing to a handful of summary numbers; in reality, there may be more sequential or rhythmic "signatures" (like how someone tends to alternate between fast bursts and short breaks) that are not captured. Representing those in a font is not straightforward, but one could imagine dynamic fonts or animations as an extreme case (though that veers into non-static typography).

Finally, we consider the broader implications of adopting personalized fonts on a large scale. If this idea catches on, digital communication could become markedly more per-

sonal, but also potentially less uniform. There may be a need for norms or standards; for example, organizations might regulate the use of individual fonts in official communication, much like a profile picture. There are also fascinating psychological questions: would seeing someone's personal font subconsciously influence how you perceive their message (e.g., attributing personality traits based on font style)? Prior work in psychology and HCI has shown that typography can affect readability and even persuasion; personalized fonts add a new layer, as they tie the typography to an individual. It's an open question whether this strengthens the emotional impact of the text (making it feel "from the person" more than plain text does) or whether most readers would not notice much difference beyond novelty. In summary, moving toward real-world usage of our system raises interdisciplinary questions spanning technology, design, user experience, and social factors. Addressing these will be key to understanding whether personalized fonts will remain a niche curiosity or evolve into a common feature of our communication landscape.

7.4 Future Improvements

While many potential extensions to this work can be envisioned, this section highlights four potential directions for future improvements. In particular, we identify the need for a more rigorous quantitative evaluation of the generated fonts, investigations into the evolution of typing behaviour over time, exploration of emotional factors in font generation, and novel applications of personal fonts in user authentication. It is hoped that the avenues discussed below will further validate and extend the capabilities of a keystroke-driven personalized font system.

7.4.1 Design Constraints and Legibility

The system generated fonts without enforcing typographic constraints related to legibility or conventional design practices. As a result, it is possible to produce outcomes such as left-leaning slants, which are uncommon in Latin typefaces and may affect readability. This reflected the project's focus on expressive personalization rather than continuous reading. For future work, the parameter space could be bound by typical typographic constraints, for example, restricting slant ranges, aligning weight with x-height, or avoiding extreme aperture values. Incorporating such considerations would allow the system to balance self-expression with legibility, making it more suitable for practical use.

7.4.2 Evolution of Typing Behaviour in the Short and Long Run

Another worthwhile direction is to investigate how a participant's typing characteristics, and hence their generated font, might change over time in both the short and long

term. Short-term variations may occur due to transient factors, such as the participant's immediate context, emotional state, fatigue, or practice effects. For example, a person might type more slowly when tired or use different rhythms when composing an email versus coding, which could potentially lead to slight shifts in their keystroke patterns. Long-term changes may occur over months or years, as individuals gain experience, adopt new devices, or as age and motor abilities gradually affect their typing. Studying these temporal dynamics would require collecting longitudinal typing data from the same participants under various conditions and at different points in time.

Analyzing the evolution of typing behaviour in this way would inform how stable or adaptable the font generation system needs to be. If a user's keystroke pattern drifts significantly over time or in different contexts, the personalized fonts produced at one point may become less representative later. This suggests that the system might benefit from a recalibration or update mechanism, such as periodically re-running font generation using recent typing data to capture new traits. On the other hand, if core aspects of a user's typing signature remain relatively stable in the long run, it would affirm that one-time font generation can reliably capture their "typing personality." Either outcome is informative. By conducting a longitudinal study of typing behaviours, future work can determine the degree of persistence in users' typing-driven font traits and design the system either to lock in a stable personal style or to flexibly adapt to gradual changes in the user's behaviour over time.

7.4.3 Fonts Reflecting Emotional Status

A further promising avenue for future work is to examine whether a user's emotional state leaves an imprint on their typing-derived fonts. This investigation would require enriching the dataset by collecting typing sessions labelled with the typist's concurrent emotional status (e.g., relaxed, stressed, happy, frustrated). Using our current feature-mapping approach, one could generate personalized fonts from typing data recorded under different reported emotions and then analyze those fonts for systematic differences. For example, do keystroke patterns typed during high-stress conditions produce fonts with noticeably different parameter values (perhaps a heavier stroke weight or more irregular shapes) compared to fonts generated from calm, steady typing? By correlating the variations in font features with the emotional context of the input data, we can assess whether the final fonts encode any latent emotional traces. In essence, this study asks, are the subtle nuances of how we type under emotional conditions reflected in the aesthetics of the fonts we generate? Finding such correlations would broaden our understanding of what personal fonts reveal and suggest that typing-driven fonts can capture not only stable behavioural traits but also situational states. Conversely, if no emotional signature is detectable in the output fonts, this is also a notable result, indicating that our mapping

may be appropriately filtering out transient states or that additional features are needed to capture the effect.

Beyond analyzing unintended emotional traces, a complementary approach is to deliberately incorporate emotional status into the font generation process. Instead of treating emotion as an uncontrolled factor, future systems could take an affective computing approach: identifying which keystroke dynamic features correlate strongly with specific emotions, and then mapping those features to font modifications designed to convey that emotion. For instance, if research finds that people typing with anger exhibit more forceful, rapid key presses, the system might map that pattern to bolder, more jagged letterforms as an "angry" font variant. Likewise, calmer typing with smooth timing might translate into more gentle curves or lighter strokes in a "calm" font. Defining a logic or set of rules for such affect-driven transformations would allow the generation of fonts that intentionally reflect the user's emotional state at the time of writing. The resulting fonts could then be evaluated in a user study for perceptual effectiveness: do observers picking up a document sense the intended emotion from the font style alone? Prior work in visual media suggests that motion or style cues can influence perceived emotion, so it is conceivable that a font that encodes emotional cues would be interpretable by readers. Pursuing this idea would merge keystroke dynamics with affective user interfaces, potentially enabling a novel mode of nonverbal expression where, for example, one's font subtly communicates mood in a written message. While ambitious, this research direction could open up new dimensions for personalized typography as a form of self-expression.

7.4.4 Personal Fonts in Authentication

Finally, personalized fonts could be explored as an auxiliary signal for verifying identity or authorship in secure communications. The premise is that a participant 's font, being derived from unique behavioural biometric data (their keystroke dynamics), carries the signature of that individual. In practice, one might envision scenarios such as document signing or email stylization, where the text is rendered in the author's personal font to provide an implicit authenticity check. For example, an official letter could be typed and displayed in the writer's custom font, much like a handwritten signature or personalized letterhead, making it immediately evident who the originator is. In multi-factor authentication systems, the consistent use of the correct personal font can complement traditional security measures (such as passwords and keystroke timing checks), adding another layer of confidence that a message or document truly came from the claimed author.

Implementing such a scheme would require rigorous control over font distribution and usage. The personal font would effectively become an identifying token, so it must not be readily obtainable or reproducible by malicious parties. If an imposter could copy or spoof someone's font file, they could falsify documents in that style, undermining

its value as an authentication factor. Therefore, any authentication application of this idea would likely involve the secure handling of the font (for instance, encrypting it for communications or verifying its use through a trusted service). Additionally, one must consider consistency. The font should render uniformly across devices and not be substitutable for look-alikes. Despite these challenges, the concept of "font-based identity verification" is an intriguing extension of personalized fonts. It underscores how fonts generated from keystroke behaviour might serve not only aesthetic or expressive purposes, but also practical ones, by reinforcing the user's identity in digital interactions. Developing secure fonts would require additional research, but if realized, this could provide a creative complement to existing biometric authentication systems.

Bibliography

[1] Shehzad Afzal, Ross Maciejewski, Yun Jang, Niklas Elmqvist, and David S. Ebert. Spatial text visualization using automatic typographic maps. IEEE Trans. Vis. Comput. Graph., 18(12):2056–2564, Dec 2012. doi: 10.1109/TVCG.2012.264. URL https://doi.org/10.1109/TVCG.2012.264.

[2] Eric Alexander, Chih-Ching Chang, Mariana Shimabukuro, Steven Franconeri, Christopher Collins, and Michael Gleicher. Perceptual biases in font size as a data encoding. IEEE Transactions on Visualization and Computer Graphics, 24(8):2397–2410, 2018. doi: 10.1109/TVCG.2017.2723397.

[3] Sofie Beier. Reading Letters: Designing for Legibility. BIS Publishers, Amsterdam, 2012.

[4] Sofie Beier and Chiron Oderkerk. Closed letter counters impair recognition. Applied Ergonomics, 101, 2022. ISSN 0003-6870. doi: 10.1016/j.apergo.2022.103709.

[5] Franco Bergadano, Daniele Gunetti, and Claudio Picardi. User authentication through keystroke dynamics. ACM Trans. Inf. Syst. Security, 5(4):367–397, 2002. doi: 10.1145/581271.581272.

[6] Michael Borenstein, Larry V. Hedges, Julian P. T. Higgins, and Hannah R. Rothstein. Introduction to Meta-Analysis. John Wiley & Sons, 2009. doi: 10.1002/9780470743386.

[7] Richard Brath. Visualizing with Text. CRC Press (Taylor & Francis), Boca Raton, FL, USA, 2020. ISBN 9780367357132. doi: 10.1201/9780429290565.

[8] Richard Brath and Ebad Banissi. Using typography to expand the design space of data visualization. She Ji: The Journal of Design, Economics, and Innovation, 2(1):59–87, 2016. doi: 10.1016/j.sheji.2016.05.003.

[9] Richard Brath and Ebad Banissi. Bertin's forgotten typographic variables and new typographic visualization. Cartography and Geographic Information Science, 46(2):119–139, 2019. doi: 10.1080/15230406.2018.1516572.

[10] Robert Bringhurst. The Elements of Typographic Style. Hartley & Marks, 2019. URL https://books.google.ca/books?id=Llcc0AEACAAJ.

[11] Tianyuan Cai, Shaun Wallace, Tina Rezvanian, Jonathan Dobres, Bernard Kerr, Samuel Berlow, Jeff Huang, Ben D. Sawyer, and Zoya Bylinskii. Personalized font recommendations: Combining ML and typographic guidelines to optimize readability. In Proceedings of the 2022 ACM Designing Interactive Systems Conference (DIS '22), New York, NY, USA, 2022. Association for Computing Machinery. doi: 10.1145/3532106.3533457.

[12] Calligraphr. Calligraphr – turn your handwriting into a font. https://www.calligraphr.com, 2025.

[13] Varun Chandola. Free-text keystroke dynamics: Latency variations and timing artifacts. https://objects.lib.uidaho.edu/. Accessed: 2025-07-03.

[14] Jacob Cohen. Statistical Power Analysis for the Behavioural Sciences. Lawrence Erlbaum Associates, 2nd edition, 1988. ISBN 9780805802832. doi: 10.4324/9780203771587.

[15] Bart Cooreman and Sofie Beier. A theory of visual attention based assessment of font style: How important is x-height for font legibility? In Society for the Scientific Study of Reading Annual Conference (SSSR '24), 2024. URL https://www.triplesr.org/. Poster session, Copenhagen, Denmark.

[16] Jane Doe and John Roe. Scalability in font design: An analysis. International Journal of Typography, 12(3):45–57, 2014.

[17] Clayton Epp, Michael Lippold, and Regan L. Mandryk. Identifying emotional states using keystroke dynamics. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11), pages 715–724, New York, NY, USA, 2011. ACM. doi: 10.1145/1978942.1979046.

[18] Brian S. Everitt. The Cambridge Dictionary of Statistics. Cambridge University Press, 2nd edition, 1998. ISBN 9780521593468.

[19] FontForge Team. Fontforge: Open source font editor. https://fontforge.org, 2023. Accessed: May 2025.

[20] Gene V. Glass. Primary, secondary, and meta-analysis of research. Educational Researcher, 5(10):3–8, 1976. doi: 10.3102/0013189X005010003.

[21] J. D. Gould and C. Lewis. Designing for usability: key principles and what designers think. Communications of the ACM, 28(3):300–311, 1985. doi: 10.1145/3166.3170.

[22] Lev Grossman. Prototypo: Design app creates bespoke typefaces. WIRED Magazine, 2014. https://www.wired.com/story/prototypo-design-app-bespoke-typefaces/.

[23] Hideaki Hayashi, Kohtaro Abe, and Seiichi Uchida. Glyphgan: Style-consistent font generation based on generative adversarial networks. Knowledge-Based Systems, 186:104927, 2019. doi: 10.1016/j.knosys.2019.104927.

[24] Larry V. Hedges. Statistical considerations for effect size standardization. Journal of Educational Statistics, 10(3):207–232, 1985. doi: 10.3102/10769986010003207.

[25] Xin Hu and Rolf D. Hersch. A parametric typeface design system. Computer Graphics Forum, 13(2):75–86, 1994. doi: 10.1111/1467-8659.1320075.

[26] Augustin-Catalin Iapa and Vladimir-Ioan Cretu. Shared data set for free-text keystroke dynamics authentication algorithms. Preprints, 2021. doi: 10.20944/preprints202105.0255.v1. URL https://doi.org/10.20944/preprints202105.0255.v1. Posted: 11 May 2021.

[27] Florian Kadner, Yannik Keller, and Constantin A. Rothkopf. AdaptiFont: Increasing individuals' reading speed with a generative font model and bayesian optimization. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21), pages 1–11, New York, NY, USA, 2021. ACM. doi: 10.1145/3411764.3445140.

[28] Leon Kadner, Sebnem Uslu, Anna Maria Feit, and Antti Oulasvirta. Adaptifont: A personalized font generator to improve reading speed. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, pages 1–14. ACM, 2022. doi: https://doi.org/10.1145/3491102.3517635.

[29] Pawel Kasprowski, Zaneta Borowska, and Katarzyna Harezlak. Biometric identification based on keystroke dynamics. Sensors, 22(9):3158, 2022. doi: 10.3390/s22093158.

[30] H. M. Kim, S. Azarm, and T. Guy. Data-driven design (d^3). Journal of Mechanical Design, 139(11):110301, 11 2017. doi: 10.1115/1.4038744.

[31] Donald E. Knuth. METAFONT: A System for Alphabet Design. Addison-Wesley, 1986.

[32] Agnieszka Kolakowska, Aleksandra Landowska, Pawel Jarmolkowicz, Michal Jarmolkowicz, and Krzysztof Sobota. Automatic recognition of males and females among web browser users based on behavioural patterns of peripherals usage. Internet Research, 26(5):1093–1111, 2016. doi: 10.1108/IntR-05-2014-0124.

[33] Mike Kuniavsky. Smart Things: Ubiquitous Computing User Experience Design: Ubiquitous Computing User Experience Design. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. ISBN 9780080954080.

[34] Johannes Lang and Miguel A. Nacenta. Perception of letter glyph parameters for infotypography. ACM Trans. Graph., 41(4), July 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530111. URL https://doi.org/10.1145/3528223.3530111.

[35] Yunhui Lin, Guoying Yang, Yuefeng Ze, Lekai Zhang, Baixi Xing, Xinya Liu, and Ruimin Lyu. The impact of motion features of hand-drawn lines on emotional expression: an experimental study. Computers & Graphics, 119:103897, 2024. ISSN 0097-8493. doi: https://doi.org/10.1016/j.cag.2024.103897. URL https://www.sciencedirect.com/science/article/pii/S0097849324000244.

[36] Tahar Mekhaznia, Chawki Djeddi, and Sobhan Sarkar. Personality Traits Identification Through Handwriting Analysis, pages 155–169. 03 2021. ISBN 978-3-030-71803-9. doi: 10.1007/978-3-030-71804-6_12.

[37] Microsoft Typography Team. Opentype font variations overview. https://learn.microsoft.com/en-us/typography/opentype/spec/otvaroverview, 2018. Accessed: 2025-06-01.

[38] Blagoj Mitrevski, Arina Rak, Julian Schnitzler, Chengkun Li, Andrii Maksai, Jesse Berent, and Claudiu Musat. Inksight: Offline-to-online handwriting conversion by learning to read and write. Transactions on Machine Learning Research, 2025. doi: 10.48550/arXiv.2402.05804. Preprint.

[39] Fabian Monrose and Aviel D. Rubin. Keystroke dynamics as a biometric for authentication. Future Generation Computer Systems, 16(4):351–359, 2000. doi: 10.1016/S0167-739X(99)00059-X.

[40] Marco Müller and Alexis Reigel. Metaflop: A web interface for parametric typeface design. Proceedings of the Typographic Research Symposium, 2013.

[41] Miguel A. Nacenta, Uta Hinrichs, and Sheelagh Carpendale. Fatfonts: Combining the symbolic and visual aspects of numbers. In Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12), pages 407–414, New York, NY, USA, 2012. ACM. doi: 10.1145/2254556.2254636.

[42] C. F. Ng. Behavioral mapping and tracking, page 29–52. John Wiley & Sons, 2016. doi: 10.1002/9781118977293.ch3.

[43] Theresa M. Nguyen, Alex D. Leow, and Olusola Ajilore. A review on smartphone keystroke dynamics as a digital biomarker for understanding neurocognitive functioning. Brain Sciences, 13(6):959, 2023. doi: 10.3390/brainsci13060959.

[44] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces, page 249–256. ACM, 1990. doi: 10.1145/97243.97281.

[45] University of Virginia. Parametric font scalable methods. Online technical report, 2011. Retrieved from UVA Computer Graphics Lab.

[46] Hilary Palmén, Michael Gilbert, and David Crossland. How bold can we be? the impact of adjusting font grade on readability in light and dark polarities. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), New York, NY, USA, 2023. Association for Computing Machinery. doi: 10.1145/3544548.3581552.

[47] A. H. Reed, R. J. Henry, and W. B. Mason. Reference values in laboratory medicine: Clinical significance of variations. Clinical Chemistry, 48(8):1181–1185, 2002.

[48] Linus Romer. mf2outline: A python script for converting metafont to outlines. https://github.com/linusromer/mf2outline, 2023. Accessed: May 2025.

[49] Rashik Shadman, Ahmed Anu Wahab, Michael Manno, Matthew Lukaszewski, Daqing Hou, and Faraz Hussain. Keystroke dynamics: Concepts, techniques, and applications. ACM Computing Surveys, 57(11):1–37, 2025. doi: 10.1145/3733103.

[50] Alice Smith and Bob Brown. Personalized fonts improve reading speed. ACM Transactions on Accessible Computing, 13(1):1–20, 2020. doi: 10.1145/3379504.

[51] Robert R. Sokal and F. James Rohlf. Biometry: The Principles and Practice of Statistics in Biological Research. W. H. Freeman, 3rd edition, 1995. ISBN 9780716724115.

[52] Yan Sun, Hayreddin Ceker, and Shambhu Upadhyaya. Shared keystroke dataset for continuous authentication. In 8th IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, UAE, 2016. IEEE. doi: 10.1109/WIFS.2016.7823894.

[53] Yuki Tatsukawa, I-Chao Shen, Anran Qi, Yuki Koyama, Takeo Igarashi, and Ariel Shamir. Fontclip: A semantic typography visual–language model for multilingual font applications. Computer Graphics Forum, 43(2):e15043, 2024. doi: 10.1111/cgf.15043.

[54] Metaflop Team. Metaflop: Parametric font generation, 2023. URL https://www.metaflop.com. Accessed: May 2025.

[55] Pin Shen Teh, Andrew Beng Jin Teoh, and Shigang Yue. A survey of keystroke dynamics biometrics. The Scientific World Journal, 2013:408280, 2013. doi: 10.1155/2013/408280.

[56] Esra Vural, Jiaju Huang, Daqing Hou, and Stephanie Schuckers. Shared research dataset to support development of keystroke authentication. In Proceedings of the IEEE 6th International Conference on Biometrics: Theory, Applications and Systems (BTAS), pages 1–6, 2014. doi: 10.1109/BTAS.2014.6996259.

[57] Shaun Wallace, Zoya Bylinskii, Jonathan Dobres, Bernard Kerr, Sam Berlow, Rick Treitman, Nirmal Kumawat, Kathleen Arpin, Jeff Huang, and Ben D. Sawyer. Towards individuated reading experiences: Different fonts increase reading speed for different individuals. ACM Transactions on Computer-Human Interaction, 29(5): 38:1–38:56, 2022. doi: 10.1145/3502222.

Appendix A

Dataset Permissions and Ethics Compliance

This appendix provides documentation related to the ethical use of datasets employed in this study, including signed agreements, public dataset declarations, and institutional research ethics approval.

A.1 Ethics Approval

The University of Victoria Human Research Ethics Board (HREB) has reviewed and approved this research under application #24-0403. A copy of the anonymized approval document is included on the following page.

The research qualifies as secondary analysis of publicly available and/or pre-existing data. In addition to these datasets, limited supplementary typing data from the author and supervisor were included with their informed consent for calibration and internal comparison purposes. No new human subject data were collected from external participants.

University of Victoria | Research Administration Information System

Human Research Ethics Anonymized Application #24-0403

A. Research team

1. Principal investigator (faculty, faculty supervising a student or post-doctoral researcher)

Principal Investigator is a faculty member, adjunct professor or sessional instructor. The annotated guidelines has more information on this.

If the project has more than one Principal Investigator (other than you) or more than one Principal Applicant, their names should be listed under section A.3 Research Team Members.

PI name

Miguel Nacenta

PI department

PI department. If more than one department, the department you are doing the research for.

Computer Science COSI

PI position

PI position at UVic

Faculty

2. Principal applicant (students & post-docs)

A Principal Applicant is an undergraduate student, graduate student or post-doctoral fellow who will be the lead researcher (for their thesis, dissertation, project, etc.) for this study. A Principal Applicant will be granted "View and edit" access by default, and will receive notifications related to the study. If the project has more than one Principal Applicant, the additional individuals should be listed under section A.3 Research Team Members.

The annotated guidelines has more information about the distinction between the Principal Applicant and the Principal Investigator.

Does this application have a principal applicant (UVic student or post-doc conducting this research for their academic degree)?

Yes

PA name

Narges Sayah Dehkordi

PA email

nargessayah@uvic.ca

PA department

Computer Science COSI

PA position

Master's student

PA phone

PA graduate secretary's email (if the principal applicant is a graduate student. Leave blank otherwise.)

cscgsec@uvic.ca

Is the principal applicant conducting this research for their academic degree at UVic?

Yes

3. Research team members

Individuals and organizations involved in conducting your research. This includes co-principal investigators, additional principal applicants, co-investigators, other UVic students, assistants (paid or unpaid), community organizations, and clients. Team members listed will have "no access" to application as a default. You cannot assign access to team members without Netlink ID. If they need a Netlink ID go to the Affiliate Identity Management System and click on the 'Sponsor' tab to start the process. Once you get the Netlink ID you have to re-enter their name and give access permission to the application.

List all current research team members (including any UVic students or research assistants who will use the received data or biological materials to fulfill UVic thesis, dissertation, or academic requirements) and assign level of access to the application. Inclusion here satisfies only UVic institutional requirements. If you grant "View and Edit" access to more than one person, be aware that the system will not notify users if and when others are making edits to the application.

DO NOT add the PI or PA to this table as that will cause technical permission issues.

Access: View and edit project 👁 View only ✉ Receive notifications $ Contribute funding

Name	Email	Role in the project	Institutional affiliation	👁	✉	$

4. Tri-Council Policy Statement (TCPS2) - Course on Research Ethics (CORE-2022) requirements

The Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS 2) provides ethics guidance that applies to all research involving human participants – including their data and/or biological materials– conducted under the auspices of an institution eligible for funding by the federal Agencies (CIHR, NSERC, SSHRC).

The online tutorial CORE-2022 (Course on Research Ethics) is an introduction to the TCPS 2 for the research community. It focuses on the TCPS 2 ethics guidance that is applicable to all research involving human participants, regardless of discipline or methodology.

All UVic research team members who intend to engage in research with human participants are required to complete the Course on Research Ethics – CORE 2022. UVic CORE-2022 FAQ will have more detailed information about this requirement.

☐ As the PI, I confirm that all UVic research team members listed in section A (A.1, A.2, A.3) have completed the CORE 2022.

B. Project information

1. Project title

Title for your research project. You may not submit two applications with the same title.

Typing Data Analysis

2. Anticipated duration for the use of the anonymized data or human biological materials

a. Approximate start date for receiving data or biological materials

The approximate start date for receiving data or biological materials for your project should take into account the time it will take to complete and submit this application form and the period of one to two weeks required for ethical review. It is a violation of University of Victoria policy to obtain and use the anonymized data before receiving HREB ethics approval.

Upon approval

b. Approximate end date for using data or biological materials

An approximate end date for using data/biological materials for your research project.

March 2025

3. Is this application linked to one that has been recently submitted to the UVic Human Research Ethics Board?

No

C. Project funding

1. Have you and/or research team members (their names must be listed under section A. Research team) applied for or been awarded funding for this project?

No

2. Will this project receive funding from the US National Institute of Health (NIH)?

No

3. If you are a faculty member and have indicated above that you have applied for external funding, have you submitted a Research Application Summary Form (RASF) via RAIS?

You must submit an RASF every time you apply for external funding. Provide explanation, if you haven't done so.

Not applicable

Comments

D. Researcher(s) qualifications

1. In light of your research methods, the nature of the research, and the characteristics of the participants, what training, qualifications, or personal experiences do the principal investigator, the principal applicant, and/or your research team members have?

E.g. research methods course, language proficiency, committee experience, training on the equipment to be used.

I have extensive experience in data analysis and data science, having worked with diverse data types such as text, image, and medical metadata to build data-driven applications. I have successfully completed a research methods course and gained further practical experience as a teaching assistant for the same course. My expertise includes applying statistical and analytical techniques to ensure the integrity and effectiveness of research outcomes. I am fluent in English, both spoken and written, enabling clear communication with participants and effective dissemination of research findings. This combination of technical skills, academic training, and language proficiency enables me to handle the complexities of this research project.

2. Tri-Council Policy Statement - TCPS2 CORE Tutorialupdated requirements.

As of March 1, 2025, all UVic research team members who intend to engage in research with human participants are required to complete the Course on Research Ethics CORE-2022. Until March 1, 2025 CORE tutorial requirements only applied to graduate students conducting research with human participants for their UVic project, thesis or dissertation. UVic CORE-2022 FAQ will have more detailed information about this requirement.

You are no longer required to complete this section if you have provided your attestation in section A.4.

The table below may list UVic graduate students on your application who were required to provide their Course on Research Ethics (CORE) tutorial certificate, prior to new requirements in place as of March 1, 2025. You are no longer required to provide Course on Research Ethics (CORE) tutorial certificates for graduate students or any other research team member if you have already provided your attestation in A.4.

Name	Email	Role in the project	CORE tutorial completion date
Narges Sayah Dehkordi	narges.sayah75@gmail.com	Principal Applicant	September 19, 2024

Supporting documents

tcps2_core_certificate.pdf *(Other, Name: tcps2_core_certificate, Version: Final Version)*; S 19, 2024

Comments

E. Data description

1. Is your research limited to receiving primary or secondary anonymized data or data sets?
Anonymized data or information means that 'the information is irrevocably stripped of direct identifiers, a code is not kept to allow further re-linkage, and risk of re-identification of individuals from remaining indirect identifiers is low or very low' (Tri-Council Policy Statement 2, p. 57). If your research involves primary or secondary data that are not anonymized, or can be linked, please check with the Human Research Ethics office immediately to ensure the correct application is submitted

Yes

2. Is your research limited to receiving primary or secondary anonymized biological materials?
Anonymized data or information means that 'the information is irrevocably stripped of direct identifiers, a code is not kept to allow further re-linkage, and risk of re-identification of individuals from remaining indirect identifiers is low or very low' (Tri-Council Policy Statement 2, p. 57). If your research involves primary or secondary data that are not anonymized, or can be linked, please check with the Human Research Ethics office immiediately to ensure the correct application is submitted

No

F. Project description

Purpose, permission, certification

1. Briefly describe the purpose of the research and your method(s) for analyzing received data/data sets or human materials

The purpose of this research is to explore how keystroke dynamics can be used to generate personalized fonts based on individual typing patterns. We are analyzing keystroke data from pre-collected datasets to identify distinguishing features between anonymous users and stable features within each anonymous user. The method involves applying statistical techniques, such as the Sum of Standardized Mean Differences (SMD) and Coefficient of Variance (CV), to process this data. The goal is to use these features to generate custom fonts that reflect the unique typing behaviour of each user.

2. Provide the name of the researcher and their university, the lab/repository or the institution from whom/which you will receive the data, data sets or biological materials
Make sure you upload any official letter, email from the researcher who collected the original data or the institution under whose jurisdiction the data was collected, confirming you have their approval for the use of data and the data you will receive will be anonymized. If you are receiving data that is not anonymized, you must contact Human Research Ethics office as soon as possible. This documentation is required.

1. University at Buffalo: The dataset was collected by Professor Shambhu Upadhyaya, who we contacted directly. I am uploading the email indicating his consent to share the anonymized data and the consent document that we signed for them.
2. Politehnica University of Timioara: The dataset is publicly available, as described in an article by Catalin Iapa and Vladimir Ioan Cretu, titled “Shared Data Set for Free-Text Keystroke Dynamics Authentication Algorithms.” I am uploading the full paper, which includes a link to the dataset.

Supporting documents

University at Buffalo Joint Multi-modal Biometric Dataset Release Agreement-processed.pdf
(Data access request/approval, Name: University at Buffalo Release Agreement, Version: only version); O 1, 2024

Re: University at Buffalo's Keystroke Dataset.pdf
(Approval from researcher or institution for use of data, Name: University at Buffalo Access Granted, Version: only version); O 1, 2024

SharedDataSetforFreeTextKeystrokeDynamicsAuthenticationAlgorithms.pdf
(Approval from researcher or institution for use of data, Name: Shared Dataset Article Timisoara, Version: only version); O 1, 2024

3. Has another Research Ethics Board (REB) reviewed and approved the study?
If the primary data collection was conducted under a jurisdiction of another Research Ethics Board and the study obtained approval, make sure you upload the Certificate of Approval issued by that REB to confirm that the data was collected in accordance with the research ethics guidelines.

No

Request, share

4. If you are receiving anonymized data from the government, you or another research team member will likely be required by government to submit a data access request (DAR) to the specific ministry or government office. Please provide details, status of the request, or upload the access request/approval.

Not applicable.

5. If you are receiving anonymized data from a health authority, hospital, organization or agency etc. you or another member of the research team may be required by the above entity to submit a request. Please provide details, status of the request, or upload the access request/approval.

Not applicable.

6. If you intend to share the received anonymized data or biological materials with third parties in the future (e.g., graduate student(s), other researchers, community organizations, First Nations band council, government etc.) who are not listed under the research team section of this application, please explain.

Not applicable.

Data obtention, protection, destruction

7. Describe the data source, how it was obtained and the format in which it was supplied. Upload an example of the data set or data collection fields.
All forms of data (e.g., databases, transcriptions) or biological materials must be supplied to you without identifiers attached. The research goal must not include the re-identification of the original individuals/donors who provided the data or biological materials. Provide a description of data source, how it was obtained and the format in which it was supplied. This documentation is required.

1. University at Buffalo: The dataset includes free-text typing and password-entry data, where users typed a fixed short phrase multiple times. The data was shared in anonymized form after direct contact with Professor Shambhu Upadhyaya. It is in CSV format and contains records like this:

NumPad7 KeyUp 63577825421428
Return KeyDown 63577825425562
Return KeyUp 63577825425671
LControlKey KeyUp 63577825443923
LShiftKey KeyDown 63577825482923
I KeyDown 63577825483812
I KeyUp 63577825483906

2. Politehnica University of Timioara: This dataset is publicly available and contains records in text format with columns for key code, press/release (0/1), and timestamp. An example from the dataset is:

16 0 95694
65 0 95826
16 1 95879
65 1 96017

Both datasets are anonymized and have no personal identifiers.

8. Explain how you will protect the data/data sets or biological materials (e.g. password protected computer, secured lab, limited access to other UVic researchers etc.) while under your use.

The data is stored in an encrypted drive partition and is used exclusively for this research. No other researchers or individuals in the lab can access the data, and it will not be shared or used for any other purpose.

9. Describe when and how you will destroy the data or biological materials received. If you will not destroy the data or materials, please explain what will be done (e.g. retain indefinitely, return a portion to the source) and explain why.

I will permanently delete the data from my computer after the completion of my degree, as I do not plan to use the dataset for further research.

G. List of uploaded documents

Review the document requirements list and the uploaded documents to ensure that you have all the applicable documents. Make sure to remove all duplicates. Upload appendices as individual documents, instead of clustering appendices under one attachments. Incomplete applications and applications with incorrectly uploaded appendices will not be reviewed. You will be notified in this case.

App. version	Section	Descriptive name	File name	Type of document
V1.0	D.	tcps2_core_certificate	tcps2_core_certificate.pdf	Other
V1.0	F.2.	University at Buffalo Release Agreement	University at Buffalo Joint Multi-modal Biometric Dataset Release Agreement-processed.pdf	Data access request /approval
V1.0	F.2.	University at Buffalo Access Granted	Re: University at Buffalo's Keystroke Dataset.pdf	Approval from researcher or institution for use of data
V1.0	F.2.	Shared Dataset Article Timisoara	SharedDataSetforFreeTextKeystrokeDynamicsAuthenticationAlgorithms.pdf	Approval from researcher or institution for use of data

H. Signatory/departmental signoff

Select the Chair/Director/Dean or their designate to sign-off on this application for submission. Once signed-off, the application will be submitted to the Human Research Ethics Board for review.

By signing-off the application, the signatory is affirming that adequate research infrastructure is available for the conduct and completion of this research project.

Signatory name

Kevin Stanley

A.2 Dataset Permissions

A.2.1 University at Buffalo Dataset

The keystroke dynamics dataset from the University at Buffalo was used in accordance with the terms outlined in their dataset release agreement. A signed copy of the dataset release form is included on the following page.

University at Buffalo

Joint Multi-modal Biometric Dataset Release Agreement

Dataset: Keystroke Free Text and Mouse Movement Coordinate Records

Introduction

The goal of this project is to develop new techniques, technology and algorithms for automatic recognition of humans. Part of this project is the effort to collect the biometric features and store them as part of a joint multi-modal biometric dataset. This biometric dataset is meant to aid researchers in their work, to develop, train, test and evaluate the human recognition algorithms.

Release of the database

Dataset records are made available to researchers other than those specifically listed on the IRB Protocol Statement, only after the receipt of a completed and signed dataset release agreement. All the records are available on a case-by-case basis. Dataset records are released via an Internet site, CD or other media (the preferred way being Internet download). All requests for the dataset are submitted to the University at Buffalo Principal Investigator. By signing this agreement, the requestor agrees to comply with the restrictions listed below. In addition, it is the responsibility of the individual executing this agreement that the data being provided be handled and used pursuant to the rules and regulations of their institution's IRB. Any failure to conform to these restrictions results in a denied access to the Joint Multimodal Biometric Dataset.

Consent

Restrictions for use of Joint Multimodal Biometric Dataset:

1. **Requests for the multi-modal biometric dataset:** All requests for the biometric dataset records are submitted to the Principal Investigator.

2. **Redistribution:** Without prior approval from Principal Investigator, the entire or part of the biometric dataset will not be further distributed, published, copied or disseminated in any way or form, either for profit or not. This refers also to further distribution of the dataset records to any other facility or organizational unit, other than the one mentioned in the request.

3. **Publication Requirements:** No biometric features captured and part of the joint multi-modal biometric dataset records will be published or released in reports, papers and other documents, until an approval in writing is obtained from the Principal investigator. Before approval, Principal Investigator will ask the subject for permission. If the Principal Investigator approves release, no captured biometric features will be used in a way that can embarrass, discomfort or anguish the original subject. If the Principal Investigator approves release, a copy of all reports, papers or any other documents that use

Joint multi-modal biometric dataset, must be submitted immediately upon release or publication to the Principal Investigator.

4. **Citation:** All documents and papers that report on research that uses the joint multi-modal biometric dataset will acknowledge the use of the dataset by including the following citation:

   Yan Sun, Hayreddin Ceker and Shambhu Upadhyaya, "Shared Keystroke Dataset for Continuous Authentication", *8th IEEE International Workshop on Information Forensics and Security, Abu Dhabi, UAE, December 2016.*

Acknowledgement

This project has been supported in part by National Science Foundation, Grant No. CNS-1314803.

_____

Name (in capitals)

Narges Sayah

_____

Signature

_____

Date

Organization and address (in capitals)

Send to University at Buffalo Principal Investigator: Professor Shambhu Upadhyaya, 329 Davis Hall, Department of Computer Science and Engineering, University at Buffalo, NY 14260-2500, or email to shambhu@buffalo.edu.

A.2.2 Clarkson University Dataset

The Clarkson dataset was obtained via email correspondence with the original authors under written permission. Although we ultimately excluded this dataset from our research due to methodological limitations (see Chapter 4), formal permission for its use was granted for academic, non-commercial research purposes. A signed copy of the dataset agreement is shown below.

Clarkson University

Joint Multi-modal Biometric Dataset Release Agreement

Dataset: Keystroke Free Text I

Introduction

The goal of this project is to develop new techniques, technology and algorithms for automatic recognition of humans. Part of this project is the effort to collect the biometric features and store them as part of a joint multi-modal biometric dataset. This biometric dataset is meant to aid researchers in their work, to develop, train, test and evaluate the human recognition algorithms.

Release of the database

Dataset records are made available to researchers other than those specifically listed on the IRB Protocol Statement, only after the receipt of a completed and signed dataset release agreement. All the records are available on a case-by-case basis. Dataset records are released via an Internet site, CD or other media. All requests for the dataset are submitted to the Clarkson University Principal Investigator. By signing this agreement the requestor agrees to comply with the restrictions listed below. In addition it is the responsibility of the individual executing this agreement that the data being provided be handled and used pursuant to the rules and regulations of their institution's IRB. Any failure to conform to these restrictions results in a denied access to the Joint Multimodal Biometric Dataset.

Consent

Restrictions for use of Joint Multimodal Biometric Dataset:

1. **Requests for the multi-modal biometric dataset:** All requests for the biometric dataset records are submitted to the Principal Investigator.

2. **Redistribution:** Without prior approval from Principal Investigator, the entire or part of the biometric dataset will not be further distributed, published, copied or disseminated in any way or form, either for profit or not. This refers also to further distribution of the dataset records to any other facility or organizational unit, other than the one mentioned in the request.

3. **Publication Requirements:** No biometric features captured and part of the joint multi-modal biometric dataset records will be published or released in reports, papers and other documents, until an approval in writing is obtained from the Principal investigator. Before approval, Principal Investigator will ask the subject for permission. If the Principal Investigator approves release, no captured biometric features will be used in a way that can embarrass, discomfort or anguish the original subject. If the Principal Investigator approves release, a copy of all reports, papers or any other documents that use Joint multi-modal biometric dataset, must be submitted immediately upon release or publication to the Principal Investigator.

4. **Citation:** All documents and papers that report on research that uses the joint multi-modal biometric dataset will acknowledge the use of the dataset by including the following citation:

Miguel Nacenta

Name (in capitals)

Signature

2023-02-07

Date

VIXI Lab, University of Victoria

Organization and address (in capitals)

Send to Clarkson University Principal Investigator: Professor Stephanie Schuckers, Department of Electrical and Computer Engineering, Clarkson University, Box 5720, Potsdam NY 13699, or email to sschucke@clarkson.edu.

A.2.3 Politehnica University of Timișoara Dataset

The Timișoara dataset was obtained from a publicly accessible source listed in the original publication [26]. The authors explicitly state that the dataset was created for research purposes and made available to the broader research community. No formal agreement was required for use. A relevant excerpt from their publication is quoted below:

> "It was created a database with typing mode from 80 users, 410,000 key events and a total time of approximately 24 hours for the acquisition of the necessary data. The data set is available at
> https://sites.google.com/view/cataliniapa/timisoara-kd-data-set."

> "We thank the 80 volunteers who responded positively and filled out the data acquisition form so that we have this complete and available data set for future research in the field of keystroke dynamics authentication." [26]