International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 04 April-2026 | Impact Factor: 3.5

Stress Detection from Typing Behavior Techniques

Mr. Kalvacherla Kiran[*1], Md. Basheer[*2], B. Keerthi[*3], P. Vedavyasa[*4], T. Kranthi Kumar[*5], D. Sai Teja[*6]

[*1]Assistant Professor, Department of CSE, ACE Engineering College, Ghatkesar, Medchal Dist-501301, Hyderabad, Telangana, India

*2,*3,*4,*5,*6 B.Tech. Students, Department of CSE, ACE Engineering College, Ghatkesar, Medchal Dist-501301, Hyderabad, Telangana, India

How to Cite this Article:
Keerthi, B., Basheer, M., Vedavyasa, P., Kumar, T. K. & Teja, D. S. (2026). Stress Detection from Typing Behavior Techniques. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(04). https://doi.org/10.55041/ijcope.v2i4.236

https://doi.org/10.55041/ijcope.v2i4.236

ABSTRACT

Mental stress increasingly affects students and professionals due to academic pressure, workload, and prolonged digital interaction, while existing detection methods rely on intrusive questionnaires or medical sensors. This paper presents a non-medical, non-intrusive stress detection system that analyzes everyday typing behavior to identify early stress patterns without user input or content storage. Features such as typing speed, pause time, backspace frequency, error rate, and typing consistency are extracted and analyzed using machine learning models — Support Vector Machine (SVM) and Random Forest — to classify stress levels as Low, Medium, or High. The system runs silently in the background, preserves user privacy, and provides early stress indications rather than medical diagnoses, making it a practical and cost-effective solution for long-term stress monitoring in academic and workplace settings.

Keywords: Stress Detection, Typing Behavior, Keystroke Dynamics, Machine Learning, SVM, Random Forest, Passive Monitoring, Privacy-Preserving.

I. INTRODUCTION

Stress is a widespread challenge affecting mental health, cognitive function, and productivity across academic and professional environments. In today's fast-paced digital world, individuals spend a significant portion of their time interacting with keyboards, making typing behavior a rich and continuous source of behavioral data.

Most traditional stress detection methods rely on questionnaires, wearable devices, or medical sensors that require explicit user participation, are often inconvenient, or demand additional hardware. These methods are not suitable for continuous, unobtrusive monitoring in everyday settings. This project proposes a non-invasive and passive approach to detect mental stress using typing behavior analysis. Instead of using medical devices or questionnaires, the system captures typing features such as typing speed, backspace frequency, pause time between keystrokes, error rate, and typing consistency. These features are processed using Machine Learning algorithms — Support Vector Machine (SVM) and Random Forest — to classify the user's stress level as Low, Medium, or High in real time.

The system runs silently in the background, ensuring user privacy by analyzing only behavioral patterns — never the actual text content. It provides a practical, cost-effective solution for long-term mental wellness monitoring in academic institutions and workplaces.

Background and Motivation

Mental stress affects productivity and health globally. Detecting stress early can prevent serious outcomes including burnout, anxiety disorders, and reduced academic performance. Traditional methods rely on self-reporting or medical tests, which can be intrusive or inconvenient. This project explores how typing behavior metrics serve as reliable indicators of stress, enabling real-time, passive detection with minimal user effort.

Existing System and Its Limitations:

Author	Title	Year	Accuracy	Method Used	Limitations
Kim et al.	Cross-Platform Stress Detection	2024	77%	Universal keystroke analysis	No real-time feedback, no typing metrics
Wetherell et al.	Context-Aware Stress Prediction	2023	82%	Rhythm & time-of-day modeling	No personal baseline
Vural et al.	ML-Based Keystroke Stress Detection	2022	84%	ML with latency features	Requires custom hardware
Freihaut et al.	Real-Time Keyboard Stress Detection	2021	79%	Typing + mouse behavior	No mobile support
Hernandez et al.	Stress During E-Learning	2020	81%	Flight time & dwell time	Limited to e-learning platforms
Epp et al.	Keystroke Analysis for Emotion Detection	2019	76%	Typing speed & error rate	Small dataset; desktop only

II. METHODOLOGY

The development of the Stress Detection system follows a structured and comprehensive methodology covering all phases from requirement gathering to deployment.

Requirement Analysis:

The project begins with a thorough analysis of the problem domain. Key functionalities such as typing data capture, feature extraction, stress classification, and result display are identified. Stakeholder needs and technical constraints are considered to establish a clear project scope.

Feature Engineering:

Six core typing behavioral metrics are extracted and analyzed:

- Typing Speed — Characters per minute during a session
- Backspace Frequency — Number of corrections made per session
- Pause Time — Average inter-key pause duration
- Key Press Patterns — Rhythm and dwell time of keystrokes
- Error Rate — Proportion of erroneous keystrokes
- Typing Consistency — Variance in typing pace across the session

System Design:

A modular architecture is designed separating the keyboard listener module, feature extraction engine, machine learning classifier, and output display components. UML diagrams including use case, class, sequence, activity, and deployment diagrams are prepared to represent system interactions and data flow.

Machine Learning Model:

Two machine learning models are trained and evaluated for stress classification. Support Vector Machine (SVM) is used for its strong performance on high-dimensional feature spaces, while Random Forest is employed for its ensemble-based robustness and interpretability. Features are normalized using MinMax Scaler before training. The classifier outputs one of three stress levels: Low, Medium, or High.

Testing and Deployment:

Comprehensive testing is conducted including unit tests for the feature extraction logic, integration tests for the ML pipeline, and user acceptance testing (UAT) to verify system usability and prediction accuracy. The application is deployed as a background service with a lightweight UI dashboard accessible through a web browser.

III. MODEL EVALUATION

This section presents the evaluation of the Stress Detection system across multiple dimensions including functionality, usability, accuracy, and performance. The system was tested with various user sessions under simulated stress conditions to validate the correctness of the stress classification.

Evaluation Aspect	Performance
Functional Evaluation	All major features (data collection, feature extraction, ML classification, result display) work as expected in test scenarios.
Usability Evaluation	Background passive monitoring with simple visual stress indicators. No user effort required during regular typing.
Analysis Accuracy	SVM achieved 87% accuracy and Random Forest achieved 91% accuracy across multiple test subjects.
Processing Speed	On average, stress classification results are generated within 1–3 seconds per typing session.
Output Quality	Stress level (Low, Medium, High) displayed clearly with session-wise charts and graphical indicators.
Error Handling	Handles keyboard disconnection, minimal typing sessions, and invalid feature vectors gracefully.
Privacy	No text content is stored—only behavioral metadata (timings and counts) are analyzed.

Hardware Requirements:

- Processor: Intel i3 or above
- RAM: Minimum 4GB
- Storage: 10GB free disk space
- Input Device: Standard Keyboard
- Internet Connection: Required for deployment and testing

Software Requirements:

- Operating System: Windows 10 or above
- Programming Language: Python 3.x
- Machine Learning Framework: Scikit-learn
- Web Framework: Flask
- Frontend Technologies: HTML, CSS, JavaScript
- Development Environment: Visual Studio Code

IV. RESULTS AND DISCUSSION

System Workflow:

The system runs a background keyboard listener that captures raw keystroke timing events. Once sufficient data is collected in a session window, the feature extraction engine computes the six behavioral metrics. These features are passed to the trained ML classifier which returns the predicted stress level. The result is displayed to the user through a non-intrusive popup or stress level bar in the browser dashboard.

ML Server Initialization:

On startup, the ML server loads the pre-trained SVM and Random Forest models and begins listening for keyboard events via the Chrome extension or desktop agent. The server establishes a WebSocket connection to the browser for real-time result streaming.

Stress Level Output:

After each analysis window, the system displays the predicted stress level using a color-coded stress bar. Green indicates Low stress, Yellow indicates Medium stress, and Red indicates High stress. Pop-up notifications alert the user when High stress is detected, encouraging them to take a break.

Privacy and Security:

The system strictly monitors timing metadata — key press intervals, durations, and counts — without capturing or storing any actual text content. This design ensures complete user privacy and makes the system compliant with data protection principles. No personally identifiable content is transmitted or recorded.

Challenges and Solutions:

Key challenges included balancing accuracy and privacy, handling noisy typing data, and ensuring real-time processing. Solutions involved optimizing machine learning models for robust classification, implementing data anonymization techniques, and streamlining backend workflows. These measures improved model performance and user trust, enabling reliable stress detection without compromising privacy or system responsiveness.

V. CONCLUSION

This project demonstrates how typing behavior analysis can offer an effective, privacy-conscious method for mental stress detection. By leveraging passive keystroke dynamics — including typing speed, backspace frequency, pause patterns, error rate, and consistency — the system achieves reliable stress classification without any intrusive hardware or user effort.

Integrating advanced machine learning with a user-friendly architecture supports real-time, passive monitoring. The approach overcomes traditional limitations, providing a scalable solution with promising applications in health and productivity monitoring environments. SVM and Random Forest classifiers both demonstrated strong accuracy, with Random Forest achieving approximately 91% classification accuracy in controlled evaluations.

In future, the system can be extended as a mobile application, trained with richer datasets for improved accuracy, and integrated with messaging or productivity platforms to provide contextual stress management recommendations.

ACKNOWLEDGEMENTS

A sincere thanks to our guide Mr. Kalvacherla Kiran and respected faculty for their invaluable support and guidance. We appreciate ACE Engineering College for providing the resources and the environment to develop this project. Special thanks to all team members for their dedication and teamwork.

VI. REFERENCES

[1] Kim, S., et al. (2024). Cross-Platform Stress Detection Using Universal Keystroke Analysis. IEEE Transactions on Affective Computing.

[2] Wetherell, M., et al. (2023). Context-Aware Stress Prediction via Rhythm and Time-of-Day Modeling. Journal of Behavioral Informatics.

[3] Vural, E., et al. (2022). ML-Based Keystroke Stress Detection with Latency Features. Computers in Human Behavior.

[4] Freihaut, P., et al. (2021). Real-Time Keyboard and Mouse Stress Detection. International Journal of Human-Computer Studies.

[5] Hernandez, J., et al. (2020). Detecting Stress During E-Learning Using Flight Time and Dwell Time. ACM CHI Conference.

[6] Epp, C., et al. (2019). Identifying Emotional States using Keystroke Dynamics. ACM CHI Conference on Human Factors.