International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 400-405, October-2023

https://doi.org/10.47001/IRJIET/2023.710053

Keystroke Dynamic Based Stress Detection System Using an Incremental Learning Based Approach for IT Professionals

[1]M. S. D. Perera, [2]S. M. D. A. R Jayathilake, [3]J. D. Ranasinghe, [4]S. V. Bartholomeusz, [5]H. M. Samadhi Chathuranga, [6]Devanshi Ganegoda

[1,2,3,4]Undergraduate, Department of Computer Science and Software Engineering, Faculty of Computing, Sri Lanka Institute of Information and Technology, Colombo, Sri Lanka

[5,6]Lecturer, Department of Information Technology, Faculty of Computing, Sri Lanka Institute of Information and Technology, Colombo, Sri Lanka

***Abstract* - This research introduces an innovative machine learning technique for real-time stress level detection based on the analysis of individual keystroke dynamics. Through the utilization of distinctive typing patterns unique to each user, our methodology incorporates incremental learning to iteratively integrate new user inputs, thereby enhancing the accuracy of the base model. A discreet Python program quietly operates in the background, collecting keystroke dynamics without disrupting the user's experience. This natural data collection approach distinguishes our work from prior studies, which often relied on specialized keyboards, manufactured stressors, or physiological sensors. At the core of our approach is the hosting of the machine learning model on a Flask server, utilizing web-based deployment for versatility and practicality. Driven by the Random Forest algorithm, our model showcases its effectiveness in real-world scenarios, offering continuous evaluation of stress levels without intrusive measures. This research introduces a distinctive dimension to stress prediction, eliminating the need for external devices or artificial stress inductions. Moreover, it underscores the vast potential of machine learning and incremental learning paradigms in crafting adaptable, user-centric systems. Looking, forward, our future endeavors aim to integrate mobile phone touch keypress dynamics with keyboard data to construct a comprehensive predictive model, further augmenting the depth of stress assessment. In conclusion, this research underscores the transformative role of technology in stress detection, advocating for unobtrusive yet robust methodologies. By seamlessly integrating into users' interactions, our approach sets the stage for a more holistic understanding of stress and opens pathways for its effective management in an increasingly technology-driven era.**

Keywords: keystroke dynamic analysis, stress, random forest, stress detection, incremental learning.

I. INTRODUCTION

Stress constitutes a significant challenge within the domain of IT professionals. Defined as the psychological and physiological response to perceived threats or challenges, this phenomenon, as expounded in [1], arises from various sources such as financial obligations, job instability, and interpersonal conflicts. Prolonged exposure to such stressors can precipitate adverse health consequences, leading to diminished productivity and a compromised quality of life. This cascade of effects may, in turn, engender waning interest in one's professional pursuits, relationships, and personal well-being.

To address this pervasive issue, a proposed system leverages machine learning algorithms to analyze keystroke dynamics for the purpose of detecting and monitoring stress levels. This innovative approach offers the prospect of early stress identification, enabling individuals to take proactive measures in managing their stress levels, thereby fostering heightened productivity and a healthier lifestyle.

While extant solutions have embraced similar methodologies, some incorporate pressure-sensitive keyboards to gauge stress levels based on keypress intensity, while others induce artificial stress through time constraints, subsequently scrutinizing the dynamic variability of keystrokes between stressed and non-stressed individuals. Additionally, research has delved into the influence of language familiarity on keypress dynamics and its correlation with stress escalation.

However, these solutions confront challenges such as limited accessibility to specialized equipment, adaptability concerns, and precision constraints. This underscores the focal point of the research problem that this component aims to address: the detection of stress levels through keystroke dynamics. Although this terrain has been explored by various researchers, uncharted territories persist. This research

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 400-405, October-2023

https://doi.org/10.47001/IRJIET/2023.710053

endeavor seeks to navigate these unexplored domains and furnish a novel solution.

Keystroke dynamics, being inherently unique to everyone, encompasses a spectrum of variations. Some individuals may type swiftly, while others adopt a more measured approach. Error frequencies and typing styles also contribute to this diversity. Consequently, a static approach proves inadequate. Instead, an adaptive approach is imperative. The component elucidated in this paper employs an incremental learning approach, refining predictions of the user's stress levels and progressively enhancing its accuracy with use. Data collection from the current user transpires seamlessly, without impeding their day-to-day tasks, and merely necessitates a specific program and a keyboard of their preference.

This paper not only focuses upon the implementation, methodology, specifications, and adaptability of the component for stress detection through keystroke dynamics but also illustrates its novelty in the domain, offering a unique and indispensable tool for monitoring and managing stress levels in users.

II. RELATED WORK

Stress is a pervasive issue with significant implications for individuals' health, potentially escalating from mild anxiety to severe outcomes like burnouts or even life-threatening conditions. Numerous experts and researchers have proposed diverse solutions to address this critical concern. While various studies have explored stress identification in different contexts, a subset of studies shares similarities with the proposed Stress Detection via Keyboard Dynamics component discussed in this paper.

In Hernandez and Teams' research [2], stress detection via Keystroke Dynamics employed a specialized keyboard with a pressure sensor [3] to discern stress through variations in keystroke pressure. This study also integrated a capacitive mouse [4] to further validate stress-induced typing pressure irregularities. However, this approach's reliance on pressure-sensitive keyboards may limit its practicality for everyday use.

An alternative approach to stress detection [5] focused on the correlation between keystroke patterns and stress levels. This study collected typing samples in controlled settings, manipulating stress levels for participants through a multitasking game and negative feedback. While promising, this artificial stress-inducing method requires refinement for practical application in everyday contexts due to its dependence on controlled environments.

Another study [6] investigated how stress affects keyboard and mouse usage dynamics, considering factors like language familiarity, differing text lengths, and time constraints. Findings indicated heightened stress levels in unfamiliar language and time-constrained scenarios, along with increased keystroke latency under stress. While insightful, this approach requires further calibration and automation for accurate stress level assessment.

In a different study [7], researchers utilized a pressure-sensitive keyboard to discern user emotions, integrating readings from keystroke pressure variations, dynamic time warping, and conventional keystroke dynamics. This approach successfully identified a spectrum of user emotions, but the exploration in terms of detecting stress requires further development.

The research [9] aimed to identify early-stage stress levels by examining keystroke patterns of participants exposed to varying stress-inducing conditions. While effective in stress variation detection, concerns lie in the unsupervised data collection process, lacking specific software for compliance and data integrity verification.

The table below illustrates the breakdown of the research gap between the proposed system and the existing research:

Table 1: Research Gap

Research	[2]	[5]	[6]	[7]	[9]	Proposed Component
Compatible with any keyboard	✗	✗	✓	✗	✓	✓
Functional without various biosensors	✗	✗	✓	✓	✓	✓
Free from artificial stress induction	✗	✗	✗	✗	✗	✓
Adaptable to everyday use	✗	✗	✗	✗	✗	✓

When examining the existing research on stress detection through keystroke dynamics, certain recurring methodologies and challenges come to light. Many of these studies employ pressure-sensitive keyboards, conduct experiments under controlled conditions, induce stress artificially, and gather typing data without direct supervision. It's noteworthy, however, that these approaches often face hurdles when it comes to practical integration into the everyday workflows of users, particularly IT professionals.

In the specific context of IT professionals, the paramount objective is to detect stress levels with minimal interference in their routine. To achieve this, the most promising approach may entail harnessing the tools and equipment already intrinsic to the daily activities of IT professionals. This

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 400-405, October-2023

https://doi.org/10.47001/IRJIET/2023.710053

strategy holds the potential to provide a seamless and non-disruptive means of monitoring stress levels in authentic, professional settings.

. III. METHODOLOGY

Keystroke dynamic based stress detection itself is a novel approach when it comes to the stress detection domain. The components predictions are derived from the keystroke dynamic related data that is fed to the machine learning model that is built into the system.

First, a previously created dataset containing the keypress dynamics and stress levels is used in training the machine learning model. This data is analyzed, preprocessed, split and fed to a machine learning model. The model is trained based on the data that was fed into it. After the training process, the trained model is hosted in a Flask server, and it is used to predict the current stress level of the user based on their keystroke dynamic data that's fed to it. The model that was trained will perform as a base model, and as the system is being used, the base model will be retrained using new datasets utilizing an incremental learning-based approach while functioning as the base model for predictions.

The approach includes a program developed using Python language which will detect and record the keystroke dynamic related data and send the gathered data to the Flask based server. From the server the trained machine learning model that is hosted in it will acquire the gathered data for generating predictions. The program is also responsible for acquiring stress level data that was derived on the same timeframe from the specific database instance and analyze and prepare that data for the incremental learning-based approach for base model retraining.

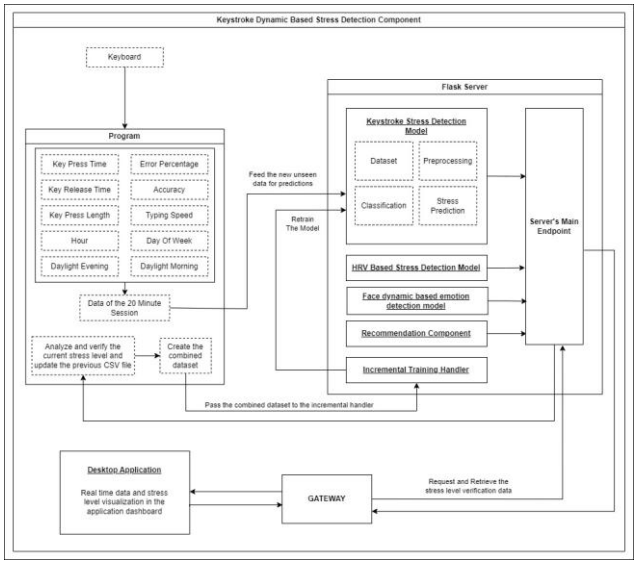


Figure 2: Component Workflow

The prediction in this approach is generated using a machine learning model (Random Forest). The initial model was trained from a currently existing dataset which contains the keypress time, release time, keypress length and stress level.

The dataset contains 12414 rows of keystroke dynamic related data and another dataset with approximately 14500 rows of keystroke dynamic related data. Both datasets were utilized in training the base models during the model selection phase of this project. During this phase the holdout validation technique was utilized to obtain a performance estimation on the model which would assist in the model selection. Before proceeding to train the models, the dataset was loaded into the google colab notebook and then the data was preprocessed and applied feature engineering techniques and tested with various approaches of these preprocessing techniques. The process included label encoding, one-hot encoding techniques, date time filtering techniques and data balancing techniques. During the data preprocessing phase, new features were derived from the dataset. They are Daylight_Evening, Daylight_Morning, Hour, Day_Of_Week.

During the model selection phase of this component various machine learning models were trained to assess the accuracy and the prediction rate and further identify what model would best suit the current scenario. The dataset contains time related data however the target variable is a categorical variable. Therefore, both possibilities were tested using two time series related models and two classification-based models.

The Auto Regressive Integrated Moving Average (ARIMA) model and the Long Short-Term Memory Model (LSTM) models were trained with the current dataset. The ARIMA Model was trained, however the prediction on the test set was not accurate since it was only able to predict only one state. Then the LSTM model was trained and was tested using the test split and it generated quite good results when trained only with the stress level variable it was able to predict the upcoming stress value, but the model was not able to predict the proper stress level when it was trained with all the features. It was also only able to predict one state. Therefore, both time series related models were removed from the selections. The SVM model training got an accuracy of 0.5954790823211876 before the data balancing techniques were used. However, since SVMs require balanced datasets to be trained the training was not accurate and the model could only predict one state. Then the model was retrained after using an oversampling method on the dataset. This produced a balanced-out dataset which resulted in increasing the accuracy to a level of 0.7777218587485322.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 400-405, October-2023

https://doi.org/10.47001/IRJIET/2023.710053

Then the Random Forest Model was trained. Which first got an accuracy of 0.593117408906882 without the feature engineering techniques but after the feature engineering was applied the model's accuracy increased up to 0.7844129554655871 and was able to predict the stress levels on the test set up to a good standard. During further tests the dataset was further preprocessed using data balancing techniques such as oversampling. The accuracy level increased up to 0.8359335681932561.

Breakdown of the classification model results during the model selection phase is as follows,

Table 1: SVM and Random Forest Accuracy Breakdown

Model Name	Accuracy	F1 Score
Support Vector Machine	0.77772186	0.77508258
Random Forest	0.83593357	0.83633066

Since both SVM and Random Forest models showed promising results, the model was chosen from the best accuracy and F1 score. Hence, the Random Forest model was chosen to perform as the base model for the component.

After the various tests that were taken to find the best feature set and the finalized dataset the final model training dataset consisted of Key_Press_Length, Stress_Level, Hour, Day_Of_Week, Daylight_Evening, Daylight_Morning as the feature set for the stress level predictions and the Table 2 displays the accuracy and F1 Score that was obtained with the use of the finalized dataset.

The base random forest model was then hosted in the Flask server allowing it to read unseen data sent to it and predict stress levels and communicate the predicted stress level to the user.

As mentioned before the unseen data relating to the keystroke dynamics is generated via a python program which can capture and derive the following dynamics from the user's session.

The program uses the Key Press time (PT) and the Release time (RT) to derive the Key Press Length (KPL). Then the Error Percentage (EP) which will be derived from the Number of backspaces or delete key presses (BS) and the Total Characters (TC). The Accuracy (A) of the entire session will be derived from the EP. Finally, the typing speed (TS) will be measured in Words Per Minute (WPM) which will take in Word Count (WC) of the session and Session Time (ST) is derived from the Cycle Start Time (CST) of the session and the Current Time (CT) of each session in Seconds (sec) and then converting them to Minutes (min) and finally deriving the WPM of the session.

Furthermore, the program will be utilizing the date and time generated from the keypress time to derive the Hour (H), Daylight Evening (DE), Daylight Morning (DM), Day of week (DOW). DOW means the day of the week from 0 to 6. Sunday = 0 and Saturday = 6 and the rest of the days get the numbers respectively.

To ensure maximum data privacy, the program employs a precautionary step by substituting alphanumeric key labels with randomized symbols. This shields the actual content from unintended exposure.

The metrics are gathered from 20-minute user interaction sessions with the system. Each session compiles data into a comma-separated value (CSV) file, meticulously logging all pertinent keypress dynamic data. After each 20-minute cycle, this compiled data is sent to the Flask server. Simultaneously, the program resets its variables, preparing for new session data. While the CSV file, critical for model retraining, is stored persistently on the client's local machine, it is scheduled for deletion within 24 hours post the completion of model retraining, providing an additional layer of protection for users' data privacy.

The Flask server reliably retrieves the sent 20-minute session data, directing it to the hosted Random Forest Model for analysis. After scrutiny, the model generates the prediction. This forecast is then routed back to the main server endpoint, making its way through the gateway to the application. Here, the stress level data is aggregated and presented. Subsequently, the compiled stress level returns to the server, preparing the persistent CSV file for model retraining. As it excludes the Stress Level variable, the aggregated stress level is appended to the CSV file. This merger with the initial dataset lays the foundation for an impeccable dataset, poised for incremental learning to fortify the base model.

A background task, scheduled every 24 hours, triggers an API call to the server, initiating the model retraining process. The procedure commences with a comprehensive test of the model's current accuracy, meticulously recorded. After retraining, the accuracy undergoes another round of evaluation. These two accuracy values are then juxtaposed. The newly retrained model is only embraced as the base model by the system if its accuracy equals or surpasses that of the existing base model. Should the system favor the new model, it promptly supplants the current base model, seamlessly taking on the mantle.

This iterative model retraining, integral to the system's ongoing use, culminates in an escalating accuracy for the base model. This, in turn, refines the precision of stress level predictions, personalized to each user. Consequently, the

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 10, pp 400-405, October-2023

https://doi.org/10.47001/IRJIET/2023.710053

system adeptly forecasts user stress levels, harnessing their keystroke dynamics, all the while refining the predictive model in stride, without impinging on users' tasks.

IV. DISCUSSION

The developed stress detection component based on keystroke dynamics marks a significant leap forward in stress assessment, fundamentally reshaping how we monitor and address stress. This innovative system seamlessly captures users' natural keyboard interactions, integrating advanced machine learning techniques. The result is a non-invasive, user-centric, and highly adaptable solution for stress detection.

In stark contrast to traditional methods, which often rely on specialized devices or controlled stress simulations, this approach leverages the everyday use of keyboards. The discreet and naturalistic data collection process ensures that users can carry on with their tasks, completely unaware of the stress assessment taking place. This safeguards against any potential bias in behavior modification.

The component excels in gathering a diverse range of keystroke dynamics, including keypress duration, press time, release time, accuracy, error rate, typing speed, and contextual cues. This comprehensive approach significantly enhances the accuracy of stress predictions by considering multiple dimensions of user behavior.

As outlined in the methodology section, the Random Forest-based machine learning model demonstrates remarkable accuracy, achieving an impressive 0.835 accuracy and an F1 score of 0.836. Its adaptability is a notable strength. With frequent retraining using new data, the model hones its predictions to align with users' evolving stress patterns. This incremental learning mechanism ensures the model remains relevant and effective over time.

Privacy and ethical considerations have been central in the component's design. The keystroke dynamics are gathered discreetly, without involving sensitive or personal information, thus safeguarding user privacy. The developed keylogger program goes a step further by replacing alphanumeric key labels with random symbols, fortifying the dataset against potential information leaks. Moreover, users can continue using their keyboards as usual, without any disruption caused by the assessment process.

The component seamlessly integrates into users' daily routines through its background operation within a Flask server. The regular acquisition of keystroke dynamics at 20-minute intervals guarantees consistent data input without impeding user activities. This adaptability across various environments, keyboards, and usage scenarios positions the component as a versatile tool for stress assessment.

In conclusion, the stress detection component based on keystroke dynamics introduces a groundbreaking and pragmatic approach to stress assessment. Its discreet nature, thorough data collection, adaptive learning model, and seamless integration set it apart as an innovative tool for comprehending and managing stress. By obviating the need for specialized hardware or conscious user engagement, this component ushers in a new era in stress detection research.

V. CONCLUSION

In conclusion, this research paper presents a pioneering approach to gauging users' stress levels through the analysis of keystroke dynamics, leveraging the distinctive patterns in individual typing styles. Powered by the Random Forest algorithm, the machine learning component provides a non-intrusive and continuous assessment of stress levels. Unlike prior studies relying on specialized keyboards, artificial stress induction, or bio sensors, this methodology taps into users' inherent typing behavior.

The incorporation of incremental learning empowers the system to adapt to new user inputs, resulting in ongoing improvement and heightened accuracy. The seamless operation of the Python program in the background ensures a user-friendly experience with uninterrupted data collection.

The deployment of the machine learning model within a Flask server demonstrates the approach's practicality in real-world scenarios. This not only offers a responsive platform for stress level prediction but also lays the groundwork for future scalability and integration into various applications.

As a future avenue of exploration, this research aims to incorporate mobile phone touchscreen key touch dynamics, adding an even more comprehensive dimension to stress assessment. By amalgamating data from multiple sources, including keyboard and mobile phone interactions, a consolidated prediction could be generated, providing a more holistic understanding of users' stress levels. This, in turn, promises a more accurate prediction of the current user's stress level by comparing and combining results gathered from both keyboard and mobile phone interactions.

In summary, this research not only introduces a distinctive perspective to stress prediction but also showcases the potential of machine learning and incremental learning techniques in crafting adaptable and user-centric systems. As technology continues to advance, the insights gleaned from this study will catalyze further innovations in the realm of stress assessment and management.

ISSN (online): 2581-3048
Volume 7, Issue 10, pp 400-405, October-2023
https://doi.org/10.47001/IRJIET/2023.710053

REFERENCES

[1] "3 types of stress and what you can do to fight them," [Online]. Available: https://www.betterup.com/blog/types-of-stress. .

[2] J. Hernandez, P. P. A. R. and M. C. , "Under Pressure: Sensing Stress of Computer Users," Association of Computing Machinery, p. 10, 2014.

[3] Paul H. Dietz, B. E. Jonathan Westhues and S. B. , "A Practical Pressure Sensitive Computer Keyboard," UIST, 2009.

[4] T. Yamauchi and K. X. , "Reading Emotion From Mouse Cursor Motions: Affective Computing Approach," Cognitive Science, 2017.

[5] L. M. Vizer, "Detecting Cognitive and Physical Stress Through Typing Behavior," 2009.

[6] S.-H. LAU, "STRESS DETECTION FOR KEYSTROKE DYNAMICS," p. 232, 2018.

[7] L. Hai-Rong, L. Z.-L. Y. W.-J. and J. D. , "Emotion recognition based on pressure sensor keyboards," IEEE, 2008.

[8] H. Lv and W.-Y. W. , "Biologic verification based on pressure sensor keyboards and classifier fusion techniques," IEEE, 2006.

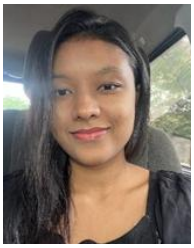
[9] A. A. M. S. Y. M. Lim, "Detecting Emotional Stress during Typing Task with," p. 10, 2014.

[10] D. Gunetti and C. P. , "Keystroke analysis of free text," ACM, 2005.

AUTHORS BIOGRAPHY

Dulshan Perera,
An undergraduate student specializing in Software Engineering at the Sri Lanka Institute of Information Technology in Colombo, Sri Lanka.

Amantha Jayathilake,
An undergraduate student specializing in Software Engineering at the Sri Lanka Institute of Information Technology in Colombo, Sri Lanka.



Janudi Ranasinghe,
An undergraduate student specializing in Data Science at the Sri Lanka Institute of Information Technology in Colombo, Sri Lanka.

Shehan Bartholomeusz,
An undergraduate student specializing in Software Engineering at the Sri Lanka Institute of Information Technology in Colombo, Sri Lanka.

Samadhi Rathnayake,
A lecturer in the Department of Information Technology at the Sri Lanka Institute of Information Technology in Colombo, Sri Lanka.

Devanshi Ganegoda,
A lecturer in the Department of Information Technology at the Sri Lanka Institute of Information Technology in Colombo, Sri Lanka.

Citation of this Article:

M. S. D. Perera, S. M. D. A. R Jayathilake, J. D. Ranasinghe, S. V. Bartholomeusz, H. M. Samadhi Chathuranga, Devanshi Ganegoda, "Keystroke Dynamic Based Stress Detection System Using an Incremental Learning Based Approach for IT Professionals" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 10, pp 400-405, October 2023. Article DOI https://doi.org/10.47001/IRJIET/2023.710053

****

