ISBN: 978-981-94-1156-6
2024 the 14th International Workshop on Computer Science and Engineering (WCSE 2024)
doi: 10.18178/wcse.2024.06.039

On the Machine Learning Based Business Process for User Identification Using Keystroke Datasets

Çağatay Menzil [1], Umut Emre Sarıkaya [1], Türkan Özdemir [1], Ezgi Yahşi [2 +] and Mustafa Keleş [2]

[1] Computer Engineering Department, Yildiz Technical University, Istanbul, Turkey
[2] Research and Development Center, Aktif Bank, Istanbul, Turkey

Abstract. One of the most important developments in authentication methods is the increasing popularity of user identification based on keystroke behaviors. To identify and verify people, this paper explores the topic of collecting and analyzing user-specific keystroke information. A complete dataset is generated by amassing unique characteristics including Hold, Down-Down, Up-Down keystroke durations, average number of keystrokes, backspace frequency, capitalization indicators, and shift key preferences. In this study, various machine learning pipelines are used to analyze the data set that contains these derived features and the associated user labels. Using these machine learning pipelines to correctly identify and attribute keyboard activities to their respective users is the main purpose. The goal of this work is to help build better user identification systems by using the nuances of keystroke properties. Results from experiments on the dataset show that the proposed machine learning pipelines successfully categorize and assign users according to their unique keyboard behaviors.

Keywords: Keystroke Behavior Analysis, User Identification Strategies, Machine Learning Based Business Process, User Behavioral Analysis

1. Introduction

User identification is crucial for various reasons, including enhancing security measures, personalizing user experiences, facilitating targeted services, and enabling efficient monitoring and analysis of user behavior for research or business purposes. The importance of using multiple identification methods, such as biometrics, passwords, and two-factor authentication, is crucial in enhancing digital security in our current technological landscape. Employing a combination of different verification techniques, such as fingerprint or facial recognition (biometrics), alongside traditional security measures like alphanumeric passwords and PINs, significantly bolsters defense mechanisms against identity theft and digital fraud. Additional layers, including two-factor authentication methods like receiving a one-time code via SMS or using a physical security token, further reinforce this security. Integrating these diverse methods not only strengthens the barrier against unauthorized access but also increases the accuracy and reliability of the identification process. Biometric data, being unique to each individual, adds a level of security that is difficult to breach, compared to more traditional methods like passwords, which can be more easily compromised. This multifaceted approach to identification is especially vital in sectors where security is paramount, such as banking, call center, healthcare, and government services, ensuring the protection of sensitive information. Keystroke dynamics play a critical role as a supplementary identification method alongside traditional and modern identification techniques. By analyzing the unique typing patterns of individuals, keystroke dynamics provide a non-intrusive and effective layer of security in identity verification processes. This approach enhances existing security measures, such as passwords and biometric systems, by offering continuous, real-time verification of a user's identity. When integrated with other identification methods, such as biometrics and two-factor authentication, keystroke dynamics contribute to a comprehensive security system. This is particularly crucial in areas requiring stringent identity verification, such as online banking, call centers and remote working environments, where ensuring the legitimacy of user identity is paramount. In recent times, the rapid advancement of technology has led to an increase in cyber security attacks and security vulnerabilities, highlighting the critical importance of

+ Corresponding author.
E-mail address: ezgi.yahsi@aktifbank.com.tr.

information security. Within this domain, user authentication systems, which serve as a fundamental component of information security, have become key areas of focus. This study specifically concentrates on identification methodologies based on a user's keystroke behavior, which is a novel approach in the evolving landscape of cyber security. By exploring the unique patterns in which individuals interact with keyboards, this method offers a distinctive and effective means of enhancing user authentication and, consequently, bolstering overall information security.

Motivation: In the conventional office setting, employees possess physical proximity to their workstations, ensuring a level of control and security over their professional tools. However, with the advent of remote work, a new set of circumstances has arisen. Call center agents, now operating from diverse locations, may find themselves in situations where they need to delegate control of the system to trusted individuals—perhaps, close associates or family members— to ensure uninterrupted workflow. Recognizing this evolving scenario, our study aims to address the inherent complexities associated with remote call center operations. By designing and implementing an advanced user identification system, we seek to not only enhance the security protocols but also distinguish between different users accessing the system. This differentiation becomes especially crucial in scenarios where the control of the system may transition between the primary employee and their designated proxy. Through this research, we aspire to contribute to the ongoing discourse surrounding cybersecurity and user authentication in remote work environments. By understanding and addressing the unique challenges faced by call center professionals in maintaining control over their systems, we aim to propose effective solutions that align with the evolving nature of contemporary work practices. Ultimately, our goal is to empower organizations and individuals alike with a robust and adaptable user identification framework tailored to the demands of remote call center operations.

Research Problem: This study proposes a novel methodology for user identification using keystroke behaviors with the help of machine learning algorithms. Therefore, we are investigating the answer to the following questions:

1. How to collect users' keystroke data and extract relevant features?
2. How can machine learning algorithms can be used to identify the users?
3. How effective and valuable are the evaluation outcomes obtained from the implemented methodology?

Contribution: In this study, we introduce a comprehensive business process for collecting, processing, and analyzing user keyboard stroke data for enhanced cybersecurity and user authentication. The process begins with the detailed collection of raw keystroke data, capturing the intricate typing behaviors of users. This data is then subjected to thorough pre-processing for cleansing and normalization, preparing it for accurate analysis. The next step involves the extraction of key features from the keystroke data, specifically selected to improve the efficacy of machine learning algorithms. These features focus on various aspects of typing patterns, such as speed, rhythm, and error rates. Following feature extraction, we train various machine learning models using this refined dataset. This phase includes optimizing the models for accuracy and robustness in identifying users based on their keystroke dynamics. Finally, the study concludes with an in-depth evaluation of these models. We assess their performance using multiple metrics to ensure their reliability and applicability in real-world scenarios. This comprehensive approach not only validates the effectiveness of our models in user authentication but also identifies potential areas for future enhancements.

Organization of the paper: After detailing our motivations and defining the research problem, we delve into an examination of relevant literature and key concepts. Subsequently, we describe the methodology used to develop our user identification system, along with the technologies employed in its implementation. Finally, we present the results of our evaluation of the proposed business process.

2. Fundamental Concepts and Literature Review

2.1. Fundamental Concepts

Naive Bayes [1] is a popular and powerful machine learning algorithm used for classification tasks, especially in text categorization, spam filtering, and recommendation systems. It's based on Bayes' theorem with a "naive" assumption of independence among features. Naive Bayes predicts the likelihood of a class using Bayes' theorem, considering the class's prior probability and the features' likelihood for that class.

Decision trees [2] are a type of machine learning algorithm used for both classification and regression tasks. They work by recursively partitioning the data into subsets based on the features that best separate the target variable. They serve as foundational elements for more advanced ensemble methods like random forests and gradient boosting, combining multiple trees to improve predictive performance and reduce overfitting. Random forest [3] is an ensemble learning method based on the construction of multiple decision trees during training and outputs the mode (classification) or average prediction (regression) of the individual trees. It is a versatile and powerful algorithm used in various fields for classification and regression tasks, known for its robustness and ability to handle high-dimensional data while minimizing overfitting.

Gradient Boosting Machines (GBMs) [4] are a powerful class of machine learning algorithms that build predictive models in the form of an ensemble of weak prediction models, typically decision trees, in a sequential manner. Gradient Boosting Machines, especially implementations like XGBoost, LightGBM, and CatBoost, have gained popularity in various machine learning com- petitions and real-world applications due to their impressive performance and versatility across regression and classification tasks. Multi-Layer Perceptron (MLP) is the most utilized model in neural network applications using the back propagation training algorithm. The definition of architecture in MLP networks is a very relevant point, as a lack of connections can make the network incapable of solving the problem of insufficient adjustable parameters, while an excess of connections may cause an over-fitting of the training data. MLPs are versatile and can be applied to various tasks, including regression, classification, pattern recognition, and function approximation. They have been widely used in machine learning and are fundamental in understanding more com- plex neural network architectures. K-Nearest Neighbors (KNN) is a popular and intuitive machine learning algorithm used for classification and regression tasks. The fundamental idea behind KNN is to predict the class or value of a data point based on the majority class or average of its k-nearest neighbors in the feature space. The choice of 'k,' representing the number of neighbors to consider, is a crucial parameter that influences the algorithm's performance. KNN is a non-parametric and lazy learning algorithm, meaning it doesn't make strong assumptions about the underlying data distribution and defers computation until a prediction is needed. Though simple and easy to understand, KNN's effectiveness heavily depends on appropriate distance metrics and preprocessing steps to handle varying feature scales. Logistic Regression is a widely used statistical method in machine learning for classification tasks. Unlike its name suggests, it is a classification algorithm rather than a regression one. The algorithm models the probability that a given input belongs to a particular class. Logistic Regression assumes a linear relationship between the features and the log-odds of the outcome. It is particularly useful when the relationship between the independent variables and the dependent variable is expected to be roughly linear, and it is robust in the presence of noise. The model is trained by optimizing the likelihood function, and it is interpretable, allowing for a clear understanding of the impact of each feature on the predicted outcome. Despite its simplicity, Logistic Regression often performs well in various real-world scenarios and serves as a foundational algorithm in the field of machine learning. Support Vector Machines (SVM) are supervised learning machines based on statistical learning theory that can be used for pattern recognition and regression. The primary goal of SVM is to find the optimal hyperplane that best separates data points into different classes. In the case of a binary classification problem, this hyperplane aims to maximize the margin, which is the distance between the hyperplane and the closest data points from each class. These closest data points are known as support vectors. SVM works by mapping input data into a high-dimensional feature space and finding the hyperplane that best divides the classes. It uses a kernel trick, which allows it to efficiently perform computations in higher dimensions without explicitly transforming the input features. SVMs can be seen as lying at the intersection of learning theory and practice. This is because an SVM can be seen as a linear algorithm in a high-dimensional space.

2.2. Literature Review

Today, digital security is becoming increasingly important. Users need to authenticate themselves securely on online platforms. However, traditional password-based authentication methods can sometimes fall short or carry security risks. Therefore, biometric authentication technologies are gaining popularity. One of these technologies is keyboard dynamics. Keystroke dynamics is a bio- metric authentication method that analyzes a person's keyboard usage habits, such as the way they press keys, the speed at which they press, and the time they hold down. This method is used in the authentication process by identifying a person's unique keyboard

usage characteristics. The way a user presses the keys is a signature unique to that person and can therefore distinguish them from other users [5]. Although important studies on user authentication with keystroke dynamics have been realized in recent years, the first studies on this subject started in the 1980s. The earliest studies on this method were performed in the years 1985-1990 for desktop keyboards. D. Umphress and G Williams are among the first to write on the subject of using keystroke patterns to identify a user. Their work explains the concept and introduces the idea of latency (also called digraphs), the amount of time between two keystrokes [6].

Various techniques have been used in analysing the keystroke dynamics of a short text sample, and an overview with a detailed analysis of many of the techniques proposed in various works can be found in [7]. One of these techniques, K-Nearest Neighbor, is a machine learning algorithm often used in keystroke dynamics analysis. The paper [8], a K-Nearest Neighbor approach has been proposed to classify users' keystroke dynamics profiles. For authentication, an input will be checked against the profiles within the cluster which has greatly reduced the verification load. In paper [9], Logistic Regression is used for user authentication with keystroke dynamics. The dataset was subjected to a pre-processing stage as in other studies and Logistic regression gave results with less accuracy than other algorithms used. The paper [7] compares the proposed techniques based on equal-error rate and zero-miss false-alarm rate. While some papers focus on exploiting a single classification technique, there are examples of using a combination of multiple methods to identify users based on keystroke dynamics in [10], a Genetic Algorithm approach is used for randomized search in combination with SVM as a base learner in the wrapper approach, while [11] uses a fusion of numerical methods with good performance properties that require a low training data volumes in the learning process and are easy to implement. Examples of user classificators based on a single classification technique include the rough set approach proposed by [12], statistical methods such as Naive Bayes classifier explored in [13] and an artificial neural network approach using auto associative Multi-Layer Perceptrons used by [14]. While the approach to collecting the training data is similar in most of the related works, approach to classification used in this work shows most similarities to the approach used by [14].

While most other approaches to classification and recognition strive to develop a mathematical model that corresponds with measured data and model a system based on the calculated model, artificial neural networks work directly with the measured data implicitly storing the model [15]. In [16] Recurrent Neural Networks (RNNs), which are a specialized type of artificial neural network architecture, was used to capture the correlations of keyboard input data in time sequence.

The authors of [17] introduces a deep neural network architecture designed to silently and continuously identify users based on their keystroke dynamics. Their approach employs a feature model to capture each user's unique typing style, achieving a precision, recall, and accuracy of 99.7%, 99%, and 99% respectively using a 9-layer MLP neural network on a merged dataset of 1530 users. Comparing their deep learning approach with traditional decision-tree algorithms highlights its superior performance in keystroke analysis.

The authors of [18] discusses that despite technological advancements, the persistent threat of impostors and fraudsters causing havoc in nations worldwide remains concerning. Instances of keyboard device breaches are on the rise, leading to authentication failures in smartphones, tablets, and laptops, even with commonly used methods like PINs and pattern drawing. Keystroke dynamics, a method studying typing behavior, offers promise in reducing these authentication issues, yet remains underutilized in these crucial devices. They provide a means to distinguish users based on their varied typing speeds.

The authors of [19] primarily focuses on the technical challenges and solutions in handling extensive datasets for improving user interface testing. However, this paper specifically zooms in on the nuanced behavior of users at the keyboard, offering a more focused insight into individual user patterns rather than large-scale data interactions. The authors of [20] provides an overview of structural code clone detection, concentrating on software metrics to identify code similarities. This work's orientation towards software development and code analysis stands in contrast to the keystroke dynamics study, which is more concerned with behavioral biometrics and personal identification, showcasing the difference in application areas and research focus. The authors of [21] discuss the QuakeSim project, which involves managing geophysical data and applications through web services. This project is a different domain compared to the personal

authentication focus of the keystroke dynamics paper. The authors of [22-23] and [25-26] explore user navigational behavior through clickstream data analysis. The authors of [27-28] discusses a data representation technique for historical provenance data. These studies are closely related to user/process behavior analysis but are primarily focused on navigational patterns and preferences of users, differing from the keystroke dynamics paper that hones in on the unique typing patterns of individuals for identification purposes. The authors of [24] discusses the infrastructure and collaboration in scientific computations, which is quite distinct from the individual-centric analysis of keystroke dynamics for user identification. Different from these previous studies, this paper is focusing on the detailed and individualized analysis of typing patterns for user identification, thereby establishing its distinctiveness in research approach and application.

3. Methodology

The pipeline shown in Figure 1 explains the methodology used to create a system for user identification using keystroke behaviors. In visual representation, rectangular shapes with black rows symbolize data sets, while blue rectangles depict modules. The elliptical blue block stand as representations of machine learning and deep learning models within this graphical depiction. We explain the details of the modules of the proposed business process below.

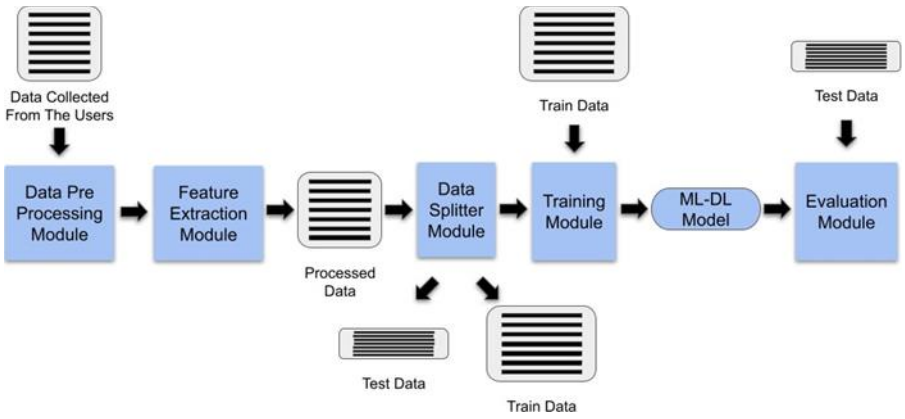

Fig. 1: The proposed business process for user identification using keystroke dynamics

Data Pre-Processing Module: Real-life datasets have some problems since they contain large amounts of data, they also contain noisy, inconsistent, and missing data. These are the factors that reduce the quality of the data set. Since the quality of the data set is the most fundamental factor affecting the accuracy of the models to be established, machine learning algorithms cannot be applied directly to every data set. Data sets need to go through a stage called data pre-processing to be freed from these problems. Before proceeding to the data pre-processing stage, the dataset used in the study is examined.

The dataset used in this study comprises a collection of data points arranged in rows and columns. Each row represents an individual observation or instance, while the columns denote specific attributes or features. The dataset is structured in a tabular form, typically stored as a text file or spreadsheet, containing various features and their corresponding values. In this specific dataset example provided, the columns encompass several features, including 'H', 'DD', 'UD', 'key stroke average', 'back space count', 'used caps', 'shift left favored', and 'label'. The 'label' column serves as the dependent feature or target variable, signifying the class or category each observation belongs to. In this context, the 'label' column appears to represent a categorical target label, potentially indicating different classes or categories. The remaining columns ('H', 'DD', 'UD', 'key stroke average', 'back space count', 'used caps', 'shift left favored') are presumably the independent features or attributes characterizing each observation within the dataset, providing numerical or categorical information relevant to the classification or analysis task at hand.

Missing data (or missing values) is defined as the data value that is not stored for a variable in the observation of interest. The problem of missing data is relatively common in almost all research and can have a significant effect on the conclusions that can be drawn from the data.

Outlier analysis, another step in data pre-processing, is a method for detecting extreme and abnormal values of attributes. These extreme values can be observed due to some external factors, exceptional

circumstances, or incorrect data entry. Therefore, in order to increase the accuracy of the model to be established, it is necessary to clear the data set from these discrete values. Outlier analysis can be applied to attributes containing numeric values. For this reason, outlier analysis was applied to all attributes except the target attribute in this dataset.

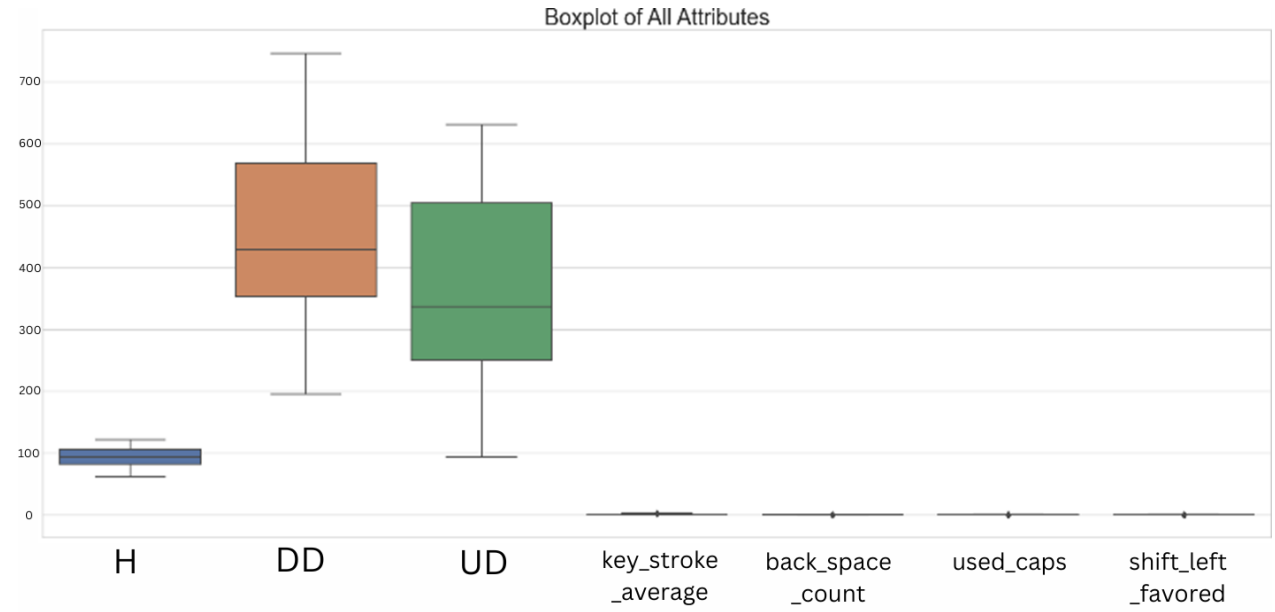

Fig. 2: Outlier analysis for all attributes with box plots

As seen in the Figure 2, point outliers were found in 7 features. These outliers were ignored since there was very few point outliers. For this reason, no action was taken to destroy the outliers. Another data pre-processing stage is data transformation. This method is used to transform these categorical attributes into numerical format, which is essential for many machine learning algorithms that require numerical inputs. In this dataset only the target attribute, which is the label attribute, is categorical, so data transformation is applied to it. Each unique string in label attribute is assigned a specific integer value. This encoding process helps in standardizing categorical data into a format that the machine learning models can interpret more effectively. However, it's important to note that the assigned integer values are ordinal and do not imply any specific ordinal relationship between the categories, as the numbers are arbitrarily assigned.

Feature Extraction Module: In this module, data gathered from user keystrokes is processed and analyzed for use in machine learning algorithms. Raw data has 4 columns which define the user who typed the text, the key which is pressed or released, the event if the key is being pressed or released and the time in milliseconds which the event occurs.

This process involves extracting several key features based on keystroke dynamics, a method crucial for assessing individual typing patterns. The features include "H," which denotes Hold time,

the average time elapsed between pressing and releasing a key. "DD," or Down-Down time, refers to the average duration from pressing one key to the pressing of the subsequent key. In contrast, "UD" or Up-Down time, measures the time between releasing a key and pressing the next one. Another significant feature is the Key Stroke Average, which calculates the average number of keys pressed within a 500ms timeframe, thereby providing an indication of typing speed. The Backspace Count is also monitored to determine the average number of backspace key presses within the same timeframe, reflecting the user's error rate or frequency of corrections. Additionally, the Used Caps feature assesses whether the caps lock was used for capitalization. Lastly, the Shift Left Favored feature indicates a preference for the left shift key over the right, offering insights into hand dominance and ergonomic typing practices. These features collectively paint a detailed picture of a user's unique typing behavior, which is invaluable for various machine learning applications.

Data Splitter Module: The data splitter module plays a crucial role, tasked with partitioning a dataset into distinct subsets designated for training, validation, and testing. Its primary objective is to guarantee that the model undergoes evaluation using data it hasn't encountered during the training process.

Training Module: The training module is responsible for fitting the various machine learning models with the training data. This module serves as the engine room where the models learn from the provided training data. Its core task involves iteratively adjusting and optimizing model parameters to capture patterns, relationships, and features within the data set. Through this process, the models aim to understand the nuances and intricacies of the data, enabling them to make accurate predictions or classifications when faced with new, unseen data.

Evaluation Module: Within the process, the evaluation module takes charge of assessing the efficacy of a trained machine learning models. Its primary objective is to offer valuable insights into the model's performance and pinpoint areas for enhancement. Leveraging the trained model, predictions are generated on the test data, and the resulting outputs are compared against the true labels. Various performance metrics, including accuracy, precision, recall and F1 score, are computed to comprehensively evaluate the model's effectiveness.

4. Prototype and Evaluation

Prototype and Evaluation Strategy: Implementation of this research utilized by Python programming language as primary. Flask framework which is programmed by Python, was employed in building the web application, while HTML and CSS were used for the front-end design. Keystroke behaviors and attributes were achieved through the use of the Pyxhook library. Machine learning algorithms are used to analyze users keystroke dynamics. Seven choices are available for user identification: Naive bayes, decision tree, random forest, gradient boosting machines, multi-layer perceptron, recurrent neural network and support vector machine.

Data Sets: Creating a custom data set involved the participation of 10 distinct users using a key logger program. Each user diligently carefully typed text 45 times, contributing to the data set's richness and diversity. Subsequently, this comprehensive data set underwent a division, separating it into distinct train and test data sets. This segmentation allows for a meticulous evaluation and validation of the models developed, ensuring their robustness and accuracy in handling new, unseen data instances.

Test Design: To assess the efficacy of the models, the dataset is split into training and test sets, with the latter exclusively utilized for evaluation purposes. The objective was to evaluate whether the models could accurately identify the correct user. For this purpose, accuracy, precision, recall, and F1-score values are calculated.

Experimental Results: The Table 1 titled "Evaluation Results" presents the performance metrics of four different machine learning methods: Naive Bayes, Decision Tree, Random Forest, and Gradient Boosting Machine. These metrics are crucial for understanding the effectiveness of each model in a classification task.

Table 1: Evaluation Results

Methods	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.86	0.88	0.86	0.85
Decision Tree	0.93	0.93	0.93	0.93
Random Forest	0.97	0.97	0.97	0.97
Gradient Boosting Machine	0.98	0.99	0.98	0.98
Multi-Layer Perceptron	0.97	0.99	0.98	0.98
K-Nearest Neighbors	0.96	0.98	0.97	0.97
Logistic Regressions	0.91	0.92	0.91	0.91
Support Vector Machine	0.98	0.97	0.97	0.98

Feature Importance: In this section, we present the feature importance as determined by three different models: Decision Tree, Random Forest, and Gradient Boosting Machine (GBM). Feature importance helps in understanding which features most significantly impact the model's decision-making process. It provides insights into the data set's structure and which attributes might be most useful for predictive modeling. Figures below show the graphical representation of feature importance for each of the respective models.

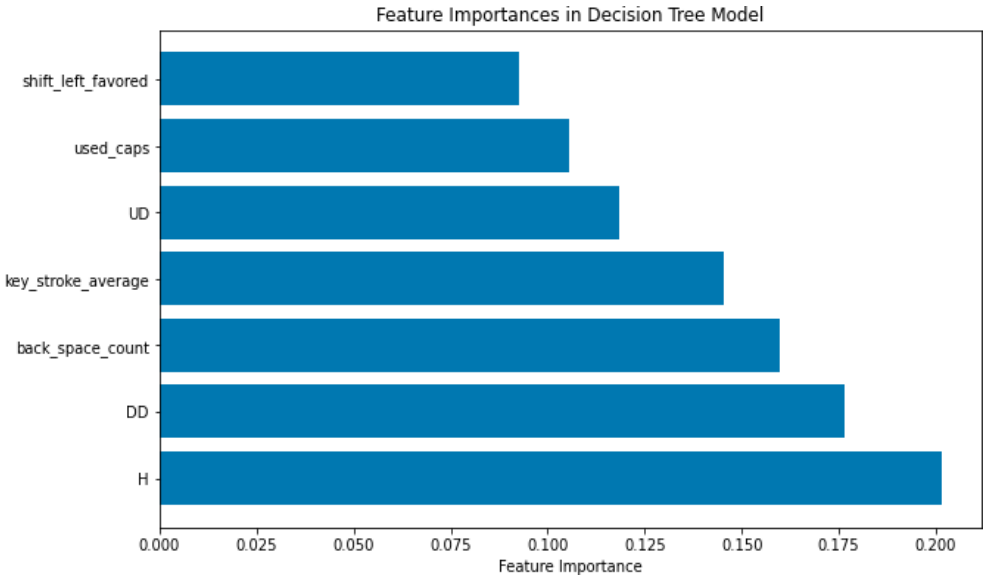

Fig. 3: Feature Importance in Decision Tree

5. Conclusion and Future Work

User identification utilizing keyboard behavior represents a cutting-edge approach in discovering and examining user-specific biometric characteristics. This method leverages the uniqueness of individuals' typing patterns as a means to enhance authentication processes. By analyzing various aspects such as typing speed, rhythm, and key press durations, it's possible to create a highly individualized profile for each user. This approach holds significant potential, especially in fields where secure access and user verification are critical like call centers and remote working environments. It serves not only as an additional security layer but also as a seamless and non-intrusive method for ensuring that the person accessing a system is indeed an authorized individual. This technology's integration into security protocols could profoundly impact industries that handle sensitive data or require stringent user verification, offering a more personalized, secure, and efficient method for user identification. Looking towards future work in this field, there is immense potential for further research and development to enhance the sophistication and applicability of keystroke dynamics. One promising direction is the integration of advanced machine learning and artificial intelligence algorithms, which can potentially offer more nuanced and adaptive analysis of typing behaviors. Additionally, exploring ways to combine keystroke dynamics with other forms of biometric verification could lead to even more secure and foolproof authentication systems. Furthermore, as our world becomes increasingly digital and security concerns grow, the appli- cation of keystroke dynamics is likely to expand into new domains. Future work might explore its use in areas such as online banking, e-commerce, and personal device security, where ensuring the legitimacy of a user is critical.

In sum, as we navigate towards a future where digital security is more crucial than ever, the role of keystroke dynamics in safeguarding our identities and information is poised to become increasingly significant. With continued research and development, this innovative approach has the potential to transform the landscape of user authentication and play a pivotal role in the security strategies of tomorrow.

6. Acknowledgement

Thanks to Aktif Bank for supporting this study by providing all the necessary requirements.

7. References

[1] Rish, I., et al.: An empirical study of the naive bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence. Volume 3. (2001) 41–46

[2] De Ville, B.: Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics **5**(6) (2013) 448–455

[3] Biau, G., Scornet, E.: A random forest guided tour. Test 25 (2016) 197–227

[4] Natekin, A., Knoll, A.: Gradient boosting machines, a tutorial. Frontiers in neurorobotics 7 (2013) 21

[5] Ivannikova, E., David, G., H¨am¨al¨ainen, T.: Anomaly detection approach to keystroke dynamics based user authentication. In: 2017 IEEE Symposium on Computers and Communications (ISCC), IEEE (2017) 885–889

[6] Mishra, V., Gupta, R., Sood, G., Patni, J.: User authentication using keystroke dynamics. Recent Trends Sci Technol Manag Soc Dev **73** (2018)

[7] Killourhy, K.S., Maxion, R.A.: Comparing anomaly-detection algorithms for keystroke dynamics. In: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, IEEE (2009) 125–134

[8] Hu, J., Gingrich, D., Sentosa, A.: A k-nearest neighbor approach for user authentication through biometric keystroke dynamics. In: 2008 ieee international conference on communications, IEEE (2008) 1556–1560

[9] Shabliy, N., Lupenko, S., Lutsyk, N., Yasniy, O., Malyshevska, O.: Keystroke dynamics analysis using machine learning methods. Applied Computer Science **17**(4) (2021) 75–83

[10] Yu, E., Cho, S.: Keystroke dynamics identity verification—its problems and practical solutions. Com- puters & Security **23**(5) (2004) 428–440

[11] Hocquet, S., Ramel, J.Y., Cardot, H.: User classification for keystroke dynamics authentication. In: Advances in Biometrics: International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007. Pro- ceedings, Springer (2007) 531–539

[12] Revett, K., Tenreiro de Magalhaes, S., Santos, H.M.: On the use of rough sets for user authentication via keystroke dynamics. In: Progress in Artificial Intelligence: 13th Portuguese Conference on Arit- ficial Intelligence, EPIA 2007, Workshops: GAIW, AIASTS, ALEA, AMITA, BAOSW, BI, CMBSB, IROBOT, MASTA, STCS, and TEMA, Guimar˜aes, Portugal, December 3-7, 2007. Proceedings 13, Springer (2007) 145–159

[13] Janakiraman, R., Sim, T.: Keystroke dynamics in a general setting. In: Advances in Biometrics: International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007. Proceedings, Springer (2007) 584–593

[14] Cho, S., Han, C., Han, D.H., Kim, H.I.: Web-based keystroke dynamics identity verification using neural network. Journal of organizational computing and electronic commerce **10**(4) (2000) 295–307

[15] Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall PTR (1998)

[16] Kobojek, P., Saeed, K.: Application of recurrent neural networks for user verification based on keystroke dynamics. Journal of telecommunications and information technology (2016)

[17] Bernardi, M.L., Cimitile, M., Martinelli, F., Mercaldo, F.: Keystroke analysis for user identification using deep neural networks. In: 2019 International Joint Conference on Neural Networks (IJCNN). (2019) 1–8

[18] Olufemi, O.G., Alimi, R.D.: Authenticating device users via keyboard strokes. International Journal of Computer Applications **975** 8887

[19] Uygun, Y., et al.: On the large-scale graph data processing for user interface testing in big data science projects. 2020 IEEE International Conference on Big Data (Big Data), 2049-2056 (2020)

[20] Kapdan, M., et al.: On the structural code clone detection problem: A survey and software metric based approach. 14th International Conference on Computational Science and Its Applications (ICCSA) (2014)

[21] Pierce, M., et al.: The quakesim project: Web services for managing geophysical data and applications. Earthquakes: Simulations, Sources and Tsunamis, 635-651 (2008)

[22] Olmezogullari, E., Aktas, M.S.: Pattern2vec: Representation of clickstream data sequences for learning user navigational behavior. Concurrency and Computation: Practice and Experience 34 (9) (2022)

[23] Olmezogullari, E., et al.: Representation of click-stream data sequences for learning user navigational behavior by using embeddings. 2020 IEEE International Conference on Big Data (Big Data), 3173-3179 (2020)

[24] Nacar, M.A., et al.: Vlab: collaborative grid services and portals to support computational material science. Concurrency and Computation: Practice and Experience 19 (12), 1717-1728 (2007).

[25] M. Oz, et al. “On the Use of Generative Deep Learning Approaches for Generating Hidden Test Scripts”, International Journal of Software Engineering and Knowledge Engineering, Vol 31, Issue 10, p1447, 2021.

[26] I. Erdem, et al. “Test Script Generation Based on Hidden Markov Models Learning From User Browsing Behaviors”, 2021 IEEE International Conference on Big Data (Big Data), IEEE Computer Society, 2021.

[27] Chen, P., Plale, B., Aktas, M.S., Temporal representation for scientific data provenance, 2012 IEEE 8th International Conference on E-Science, e-Science 2012, 2012, 6404477

[28] Aktas, M., Plale, B., Leake, D., Mukhi, N., Unmanaged workflows: Their provenance and use, Studies in Computational Intelligence, 2013, 426, pp. 59–81, 2013.