

User Behavior Analysis Using Web-based Machine Learning Features: New Solutions for IT Business

Roman Mysiuk¹, Oleksii Kononenko², Andriy Svystovych², Oleksii Ozhyhov², Nazar Osadets², Yuriy Kuchmak², Andrii Pohrebniak², Yuriy Honsor²

¹ Ivan Franko National University of Lviv

1 Universytetska Street, Lviv, Ukraine, 79000

² Lviv University of Business and Law

99 Kulparkivska Street, Lviv, 79021, Ukraine

DOI: [10.22178/pos.104-5](https://doi.org/10.22178/pos.104-5)

JEL Classification:

D11, D83, M29, M49

Received 27.04.2024

Accepted 25.05.2024

Published online 31.05.2024

Corresponding Author:

Roman Mysiuk

mysiuk@ukr.net

© 2024 The Authors. This article is licensed under a Creative Commons Attribution 4.0 License 

Abstract. The development of information technologies in IT business increases the interest in executing machine learning models directly on the client browser, reducing the load on the server and the number of levels of access to it. At the same time, some features have advantages and disadvantages, associated with a smaller amount of information transmitted over the network, limited power of client devices, and others. Among modern client-side tools with machine learning capabilities, Tensorflow.js is suitable, which can be used to analyse user behaviour in web applications for classification and clustering models based on their behavioural patterns, predict future user behaviour trends, detect unusual or suspicious user actions, recommendation models based on their previous behaviour. The article analyses the features of implementation and the limitations associated with the use, specifically regarding the behaviour of users in social networks. The model was formed based on data from news posts on social networks Instagram and Facebook, with the following parameters of user activity, such as the number of likes, comments, and shares according to the post's text. These aspects are a significant addition to the tools that can be applied within the economic, technical, and other means of IT business development. Considering this, it is advisable to study the formation and development of the innovation management system in e-business in the future.

Keywords: business; IT business; machine learning; tensorflow; user behaviour analysis; data analysis; social network; data processing; e-business development.

INTRODUCTION

In today's world, analysing user behaviour helps to understand the needs and preferences of users, develop a personal algorithm for selecting content in social networks, improve developed products and offer new services. In addition, the analysis helps to ensure Internet security and prevent fraud. Thus, a new term, User Behavior Analytics (UBA), appeared worldwide.

User Behavior Analytics (UBA) is a process of tracking, analysing, and interpreting user actions and interactions within a digital environment, such as a website or application. It involves collecting data on user behaviour, including actions such as clicks, page views, form submissions, and transactions, and using analytical techniques to derive insights into user

preferences, patterns, and trends in the same way as it is described in [1, 2]. By understanding how users navigate and engage with digital platforms, organisations can optimise user experiences, enhance product offerings, and achieve business objectives more effectively. UBA encompasses various methods and tools, including data collection through web analytics platforms, segmentation techniques to group users based on common characteristics, and predictive analytics to anticipate future user behaviour [3, 4, 5, 6]. Ultimately, UBA enables organisations to make data-driven decisions, improve user satisfaction, and drive business growth in the digital realm.

Tensorflow.js tool provides an API for developing machine learning models in JavaScript that runs on the web. This feature makes it available to many web developers and can be used for various

tasks. Today's most common example is training and running machine learning models directly in browsers or on the server using Node.js. Servers usually have more resources to handle such tasks since models are stored and executed directly on the client side. When calculating on the client side, special attention should be paid to critical parameters, which can be considered as they are in articles [7, 8, 9].

Therefore, this can result in less data that needs to be transferred over the network. This can be a significant advantage for large models as it reduces loading time. Some aspects involve running the models directly on the client device, which can help keep the data private as it never leaves the user's device. Disadvantages include the limited power of client devices, which can lead to limitations in the size and complexity of models that can be used.

The *article aims* to improve theoretical and practical aspects of the IT business, which relate to user behaviour analysis using web-based machine learning features.

RESULTS AND DISCUSSION

Methods and features of implementation of web-based machine learning. Implementing web-based machine learning involves several methods and features to deploy machine learning models on web platforms effectively. One approach is to leverage frameworks like TensorFlow.js or PyTorch.js to deploy models directly in the browser, enabling client-side inference. Additionally, using RESTful APIs or GraphQL endpoints allows seamless integration of machine learning functionalities into web applications, facilitating communication between frontend and backend systems. Feature-wise, scalability, security, and latency optimisation are crucial considerations. Implementing techniques like model quantisation, caching, and asynchronous processing can enhance performance and user experience, ensuring smooth operation even with large-scale deployments. Furthermore, incorporating user feedback mechanisms and continuous model monitoring enables iterative improvement and maintenance of deployed machine learning systems.

A. Web-Based Methods of User Behavior Analytics. Web-based methods of user behaviour analytics encompass a range of techniques aimed at

understanding and optimising user interactions with web applications. These methods leverage various data sources, tools, and analytical approaches to gather insights into user behaviour patterns, preferences, and engagement metrics. One fundamental aspect of web-based user behaviour analytics is the collection of data related to user interactions with the website or application. This data can include page views, clicks, scroll depth, form submissions, time spent on a page, and navigation paths. Tools like Google Analytics, Adobe Analytics, or custom event-tracking solutions are commonly used to gather this data.

Once data is collected, it needs to be processed and analysed to derive actionable insights. This involves data cleaning, aggregation, and visualisation techniques to uncover patterns and trends within the user behaviour data. Visualisation tools like Tableau, Power BI, or custom dashboards can help present the data in a meaningful and digestible format. Segmentation is a crucial aspect of user behaviour analytics, where users are grouped into segments based on common characteristics or behaviours. Segmentation can be based on demographic information, geographic location, device type, referral source, or user actions. By segmenting users, organisations can better understand the needs and preferences of different user groups and tailor their web experiences accordingly. Behavioural analysis techniques, such as funnel and cohort analyses, track and optimise user journeys through the website or application. Funnel analysis involves mapping out the steps users take to complete a specific goal, such as making a purchase or signing up for a newsletter and identifying drop-off points where users abandon the process. Cohort analysis involves grouping users based on shared characteristics (e.g., sign-up date) and tracking their behaviour over time to identify trends and patterns. Machine learning and predictive analytics can further enhance user behaviour analytics by identifying patterns and making predictions based on historical data [10]. For example, machine learning algorithms can predict user churn, recommend personalised content or products, detect anomalies or fraudulent activities, and optimise conversion rates.

Real-time analytics enables organisations to monitor user behaviour and respond promptly to emerging trends or issues [11, 12]. Real-time analytics platforms like Google Analytics Real-

Time or custom-built solutions allow organisations to track user interactions as they happen and make data-driven decisions on the fly. A/B testing, or split testing, is a common technique used in user behaviour analytics to compare the performance of different variations of a web page or feature. By randomly assigning users to other variations and measuring key metrics such as conversion rate or click-through rate, organisations can determine which version performs better and iterate accordingly. Privacy and data security are paramount concerns in web-based user behaviour analytics, particularly with the increasing focus on data protection regulations such as the General Data Protection Regulation and the California Consumer Privacy Act. Organisations must ensure that they collect and process user data compliant and ethically, with appropriate safeguards in place to protect user privacy [13–16]. In conclusion, web-based methods of user behaviour analytics encompass a diverse set of techniques and tools aimed at understanding, analysing, and optimising user interactions with web applications. By leveraging data-driven insights, organisations can improve user experiences, drive engagement, and achieve their business objectives effectively.

In Figure 1, the example of code for training model and testing with a count of likes, comments, and shares in the same way as it is described in [17].

```

1  const tf = require('@tensorflow/tfjs-node');
2  const fs = require('fs');
3
4  // Stage 1. Loading and preparation data
5  const data = JSON.parse(fs.readFileSync('data.json'));
6  const inputData = tf.tensor2d(data.input);
7  const outputData = tf.tensor2d(data.output);
8
9  // Stage 2. Create and compile model
10 const model = tf.sequential();
11 model.add(tf.layers.dense({ units: 64, inputShape: [3], activation: 'relu' }));
12 model.add(tf.layers.dense({ units: 32, activation: 'relu' }));
13 model.add(tf.layers.dense({ units: 1, activation: 'sigmoid' }));
14 model.compile({ optimizer: 'adam', loss: 'binaryCrossentropy', metrics: ['accuracy'] });
15
16 // Stage 3. Model training
17 async function trainModel() {
18   await model.fit(inputData, outputData, { epochs: 50 });
19   console.log('Trained model. Done');
20 }
21
22 // Stage 4. Testing on real values
23 trainModel().then(() => {
24   // Classification testing base on count of likes, comments and shares
25   const newData = tf.tensor2d([[10, 5, 20]]);
26   const prediction = model.predict(newData);
27   console.log('Probability of popular topic:', prediction.dataSync()[0]);
28 });

```

Figure 1 – An example of code for training model and testing with a count of likes, comments and shares

Step 1. The data is loaded from the "data.json" file, where the input contains data on the number of likes, comments, and shares for each text, and the output includes labels that indicate the text's interest. This can be done using standard

programming language tools or a library to work with JavaScript Object Notation (JSON) data.

Step 2. A neural network is created from several layers, which are used to analyse the input data and determine the interest of the text. In the case of text interest classification, dense layers are usually used. For example, "tf.sequential()" creates a sequential model.

Step 3. The model is compiled using the Adam optimiser and the binary cross-entropy loss function since this is a binary classification problem. The Adam optimiser is a popular choice for many neural network training tasks because it usually helps speed up the training process and improve model convergence. It combines two other optimisation methods: Stochastic Gradient Descent (SGD) and Adaptive Learning Rate (AdaGrad) method. For binary classification problems where each example can belong to only one of two classes (e.g. "positive" or "negative", "true" or "false"), the binary cross-entropy loss function is often used because it is well suited for estimating the difference between predicted and actual values in such problems.

Step 4. The model is trained on input and output data. The "fit" method is used for this, which adapts the model parameters to the training data.

Step 5. After training, the model can be used to predict the interestingness of new text by providing input to the expected method input. The "predict" method returns probability vectors indicating the probability of each input sample belonging to the classes.

Each row in the input list represents the input characters for each text, where the first element corresponds to the number of likes, the second to the number of comments, and the third to the number of shares. Each item in the output list indicates whether the text is intriguing (1) or not interesting (0) [17].

The model example for training can be set similarly to that in Figure 2.

B. Features of Implementation Tensorflow.js.

Tensorflow.js offers cross-platform compatibility for developing machine learning applications on web browsers and Node.js, enabling client-side inference for faster response times. It facilitates model deployment by converting trained models from Tensorflow or Keras into web-compatible formats. With support for transfer learning and customisation, developers can efficiently fine-

tune models and experiment with different architectures. Tensorflow.js seamlessly integrates with web technologies like HTML and WebGL, optimising performance through hardware acceleration. Its security features, including model encryption, protect sensitive data, while continuous updates and a vibrant community contribute to its ongoing development and improvement.

```

1  {
2    "input": [
3      [10, 5, 20],
4      [8, 3, 15],
5      [15, 7, 25],
6      [5, 2, 10],
7      [20, 10, 30]
8    ],
9    "output": [
10     [1],
11     [0],
12     [1],
13     [0],
14     [1]
15   ]
16 }
    
```

Figure 2 – Part of training model example in data.json

Some details regarding the specifics of using web-based machine learning can be highlighted. Tensorflow.js allows you to run machine learning models directly in the user's browser. This can provide fast and efficient user behaviour analytics without sending data to the server. Tensorflow.js can be easily integrated with web applications, allowing developers to create user behaviour analysis tools. Using WebGL and other technologies, Tensorflow.js can handle large amounts of data and complex models, making it suitable for extensive web services.

At the same time, there are some limitations to using this approach, which are related to the limited client resources. Sometimes, the amount of data may be so large that processing it on the client may be inefficient or impossible due to limited device resources. Since machine learning models run on the client device, this can raise questions about user data privacy, mainly if the data is analysed directly on the device.

The prospects of such an approach are associated with some valuable features for data analysis. Tensorflow.js can be the basis for building new tools for analysing user behaviour directly in the browser, providing new opportunities for analytics and personalisation of web applications. Local data processing and analysis approaches can increase data-intensive web applications'

efficiency and speed. Analysing user behaviour directly on the user's device can help keep their data private because the data never leaves the device.

Comparison of client-side and server-side implementation Tensorflow. Comparisons between client-side and server-side implementations of Tensorflow include differences in the execution environment, hardware, privacy, scalability, security, network usage, availability and some aspects described in [18–22] as it is in Table 1.

Table 1 – Comparing performance and validity metrics for web-based and server-based machine learning

Type of implementation of Tensorflow with different aspects		
Aspect	Client-Side	Server-Side
Location of Execution	Runs in the client's browser or device	Runs on remote servers
Latency	Relies on the client's hardware, typically less powerful	Can leverage powerful hardware on the server
Scalability	Limited by the client's hardware and browser capabilities	Highly scalable, as it can utilise cloud computing resources
Network Usage	Requires downloading model and inference code to the client, increasing initial load time and network usage	Requires sending data to the server for processing, potentially increasing network usage during inference.
Real-time Inference	Limited by the client's hardware capabilities and network speed	Can achieve real-time inference depending on server performance and network conditions

Type of implementation of Tensorflow with different aspects		
Privacy Concerns	Data remains on the client's device, potentially reducing privacy concerns	Data is transmitted to and processed on external servers, raising privacy concerns
Hardware Requirements	Relies on the client's hardware, typically less powerful	Can leverage powerful hardware on the server.

Many different indicators can be used to evaluate efficiency and reliability. For example, execution speed is determined by the time required to perform client and server device operations. The resource usage indicator estimates CPU, memory, and other resource usage on the client and server devices. Another indicator is scalability, which allows the system to expand to handle larger volumes of data or a higher number of requests.

The security score can be evaluated based on the security measures applied to the Tensorflow client and server implementation [23, 24, 25, 26]. Estimate the data transfer between the client and server environments during model training or inference.

CONCLUSIONS

Based on the information provided, it can be concluded that there is a growing interest in executing machine learning models directly on client browsers due to the development of

information technologies in the IT business. This approach offers benefits such as reducing the load on servers and minimising the number of network access levels.

Tensorflow.js emerges as a suitable tool among modern client-side machine learning capabilities, enabling the analysis of user behaviour in web applications. However, there are both advantages and disadvantages associated with client-side execution. On the positive side, there's a reduction in network traffic due to processing data locally, and it offers the potential for real-time analysis without the need for constant server requests. Nonetheless, there are limitations, such as the restricted processing power of client devices, which can affect the complexity and efficiency of models. In analysing user behaviour in social networks, Tensorflow.js presents opportunities for building classification and clustering models based on behavioural patterns. It can aid in predicting future trends in user behaviour, detecting unusual actions, and providing personalised recommendations based on past behaviour.

The article highlights the features and limitations of implementing machine learning models for analysing user behaviour on social networks, mainly focusing on data from Instagram and Facebook posts. A model can be trained to gain insights into user engagement and preferences within these platforms by considering parameters like the number of likes, comments, and shares. These aspects are a significant addition to the tools that can be applied within the economic, technical, and other means of IT business development. Considering this, it is advisable to study the formation and development of the innovation management system in e-business in the future.

REFERENCES

1. G. Martín, A., Fernández-Isabel, A., Martín de Diego, I., & Beltrán, M. (2021). A survey for user behavior analysis based on machine learning techniques: current models and applications. *Applied Intelligence*, 51(8), 6029–6055. doi: [10.1007/s10489-020-02160-x](https://doi.org/10.1007/s10489-020-02160-x)
2. Callara, M., & Wira, P. (2018). User Behavior Analysis with Machine Learning Techniques in Cloud Computing Architectures. *2018 International Conference on Applied Smart Systems (ICASS)*. doi: [10.1109/icass.2018.8651961](https://doi.org/10.1109/icass.2018.8651961)

3. Moon, J., Kim, Y., & Rho, S. (2022). User Behavior Analytics with Machine Learning for Household Electricity Demand Forecasting. *2022 International Conference on Platform Technology and Service (PlatCon)*. doi: [10.1109/platcon55845.2022.9932037](https://doi.org/10.1109/platcon55845.2022.9932037)
4. Ranjan, R., & Kumar, S. S. (2022). User behaviour analysis using data analytics and machine learning to predict malicious user versus legitimate user. *High-Confidence Computing*, 2(1), 100034. doi: [10.1016/j.hcc.2021.100034](https://doi.org/10.1016/j.hcc.2021.100034)
5. Kniaz, S., Brych, V., Heorhiadi, N., Tyrkalo, Y., Luchko, H., & Skrynkovskyy, R. (2023). Data Processing Technology in Choosing the Optimal Management Decision System. *2023 13th International Conference on Advanced Computer Information Technologies (ACIT), Wrocław, Poland*, 372–375. [10.1109/acit58437.2023.10275581](https://doi.org/10.1109/acit58437.2023.10275581)
6. Skrynkovskyy, R., Pavlenchyk, N., Tsyuh, S., Zanevskyy, I., & Pavlenchyk, A. (2022). Economic-mathematical model of enterprise profit maximisation in the system of sustainable development values. *Agricultural and Resource Economics: International Scientific E-Journal*, 8(4), 188–214. [10.51599/are.2022.08.04.09](https://doi.org/10.51599/are.2022.08.04.09)
7. Yuzevych V., Klyuvak O., Skrynkovskyy R. (2016). Diagnostics of the system of interaction between the government and business in terms of public e-procurement. *Economic Annals-XXI*, 160(7–8), 39–44. doi: [10.21003/ea.v160-08](https://doi.org/10.21003/ea.v160-08)
8. Mysiuk, R. V., Yuzevych, V. M., Yasynskiy, M. F., Kniaz, S. V., Duriagina, Z. A., & Kulyk, V. V. (2022). Determination of conditions for loss of bearing capacity of underground ammonia pipelines based on the monitoring data and flexible search algorithms. *Archives of Materials Science and Engineering*, 115(1), 13–20. doi: [10.5604/01.3001.0016.0671](https://doi.org/10.5604/01.3001.0016.0671)
9. Mysiuk, R., Yuzevych, V., Koman, B., & Yasynskiy, M. (2022). High Availability System for Monitoring Material Degradation Processes at the Concrete-polymer Interface. *2022 12th International Conference on Advanced Computer Information Technologies (ACIT)*. doi: [10.1109/acit54803.2022.9913086](https://doi.org/10.1109/acit54803.2022.9913086)
10. Pavlyshenko, B. M. (2021). Forming Predictive Features of Tweets for Decision-Making Support. *Lecture Notes on Data Engineering and Communications Technologies*, 479–490. doi: [10.1007/978-3-030-82014-5_32](https://doi.org/10.1007/978-3-030-82014-5_32)
11. Mysiuk, R., Yuzevych, V., Mysiuk, I., Tyrkalo, Y., Pavlenchyk, A., & Dalyk, V. (2023). Detection of Surface Defects Inside Concrete Pipelines Using Trained Model on JetRacer Kit. *2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT)*. doi: [10.1109/ELIT61488.2023.10310691](https://doi.org/10.1109/ELIT61488.2023.10310691)
12. Dzhala, R., Yuzevych, V., Mysiuk, R., Brych, V., Skrynkovskyy, R., Lozovan, V., & Tyrkalo, Y. (2022). *Simulation of Corrosion Fracture of Nano-Concrete at the Interface with Reinforcement Taking into Account Temperature Change*. Retrieved from <https://ceur-ws.org/Vol-3312/paper10.pdf>
13. Skrynkovskyy, R., Kataiev, A., Zaiats, O., Andrushchenko, H., & Popova, N. (2021). Competitiveness of The Company on The Market: Analytical Method of Assessment and The Phenomenon of The Impact of Corruption in Ukraine. *Journal of Optimization in Industrial Engineering*, 14(Special Issue), 79–86. doi: [10.22094/joie.2020.677836](https://doi.org/10.22094/joie.2020.677836)
14. Sumets, A., Kniaz, S., Heorhiadi, N., Skrynkovskyy, R., & Matsuk, V. (2022). Methodological toolkit for assessing the level of stability of agricultural enterprises. *Agricultural and Resource Economics: International Scientific E-Journal*, 8(1), 235–255. doi: [10.51599/are.2022.08.01.12](https://doi.org/10.51599/are.2022.08.01.12)
15. Skrynkovskyy, R. M. (2011). [Methodical approaches to economic estimation of investment attractiveness of machine-building enterprises for portfolio investors](#). *Actual Problems of Economics*, 118(4), 177–186.
16. Skrynkovskyy, R. (2008). Investment attractiveness evaluation technique for machine-building enterprises. *Actual Problems of Economics*, 7(85), 228–240.

17. Mysiuk, I., Mysiuk, R., Shuvar, R., Yuzevych, V., Hudyma, V., & Vizniak, Y. (2023). Category Classification of Content from Instagram Business Pages. *2023 13th International Conference on Advanced Computer Information Technologies (ACIT)*. doi: [10.1109/acit58437.2023.10275458](https://doi.org/10.1109/acit58437.2023.10275458)
18. Serniak, I., Serniak, O., Mykhailyshyn, L., Skrynkovskyy, R., & Kasian, S. (2021). Evaluation of the level of the usage of social instruments for human resource management: example of agro-processing enterprises of Ukraine. *Agricultural and Resource Economics: International Scientific E-Journal*, 7(4), 82–99. doi: [10.51599/are.2021.07.04.05](https://doi.org/10.51599/are.2021.07.04.05)
19. Jain, A. K., Sahoo, S. R., & Kaubiyal, J. (2021). Online social networks security and privacy: comprehensive review and analysis. *Complex & Intelligent Systems*, 7(5), 2157–2177. doi: [10.1007/s40747-021-00409-7](https://doi.org/10.1007/s40747-021-00409-7)
20. Sumets, A., Serbov, M., Skrynkovskyy, R., Faldyna, V., & Satusheva, K. (2020). Analysis of influencing factors on the development of agricultural enterprises based on e-commerce technologies. *Agricultural and Resource Economics: International Scientific E-Journal*, 6(4), 211–231. doi: [10.51599/are.2020.06.04.11](https://doi.org/10.51599/are.2020.06.04.11)
21. Popova, N., Kataiev, A., Nevertii, A., Kryvoruchko, O., & Skrynkovskyy, R. (2021). Marketing Aspects of Innovative Development of Business Organizations in the Sphere of Production, Trade, Transport, and Logistics in VUCA Conditions. *Studies of Applied Economics*, 38(4). doi: [10.25115/eea.v38i4.3962](https://doi.org/10.25115/eea.v38i4.3962)
22. Popova, N., Kataiev, A., Skrynkovskyy, R., & Nevertii, A. (2019). Development of trust marketing in the digital society. *Economic Annals-XXI*, 176(3–4), 13–25. doi: [10.21003/ea.v176-02](https://doi.org/10.21003/ea.v176-02)
23. Popivniak, Y. M. (2019). Cybersecurity and Protection of Accounting Data under Conditions of Modern Information Technology. *Business Inform*, 8(499), 150–157. doi: [10.32983/2222-4459-2019-8-150-157](https://doi.org/10.32983/2222-4459-2019-8-150-157)
24. Bendovschi, A. (2015). Cyber-Attacks – Trends, Patterns and Security Countermeasures. *Procedia Economics and Finance*, 28, 24–31. doi: [10.1016/s2212-5671\(15\)01077-1](https://doi.org/10.1016/s2212-5671(15)01077-1)
25. Kniaz, S., Heorhiadi, N., Sopilnyk, L., Konovalyuk, I., Tyrkalo, Y., Skrynkovskyy, R., Moroz, S., Kalashnyk, O., Khmyz, M., & Kaydrovych, K. (2021). Analysis Algorithm And Factors Of International Economic Activity In The Coordinate System Of Enterprises' Organizational Development. *Proceedings of the 38th International Business Information Management Association (IBIMA)*. 3–4 November 2021, Seville, Spain, 923–931.
26. Skrynkovskyy, R., Pawlowski, G., Harasym, P., & Koropetskyi, O. (2017). Cybernetic Security and Business Intelligence in the System of Diagnostics of Economic Security of the Enterprise. *Path of Science*, 3(10), 5001–5009. doi: [10.22178/pos.27-6](https://doi.org/10.22178/pos.27-6)