Online Feature Vector Restoration in Data Stream Mining Tasks*

Volodymyr Kartashov[1], Yevgeniy Bodyanskiy[1], Nonna Kulishova[1] and Yehor Kulishov[1]

[1] *Kharkiv National University of Radio Electronics, 14, Nauky, Ave., Kharkiv, 61166, Ukraine*

Abstract

The paper considers the problem of forming a feature space for data classification in the context of stream processing. It is shown that the quality of feature extraction directly affects the efficiency of classification algorithms, especially with limited data and high dimensionality of the feature space. A method for forming an extended feature vector based on recurrent estimates of the mean, variance, and autocorrelation of successive data points is proposed. This approach ensures adaptability to changing statistical properties of the stream and allows forming compact but informative feature vectors with low computational complexity. Experiments were conducted on the problem of classifying military objects based on images that included eight categories of equipment and personnel. Comparison of two series of experiments (using only pixels and using an extended feature vector) showed an increase in recognition accuracy by 1–3% when using the proposed method, which is most noticeable for optimized neural networks and decision trees. The optimized ensemble of classifiers demonstrated the highest accuracy (75.5%). It is noted that an extended set of features increases the resource intensity of the models, reducing the speed of predictions, which requires a compromise between quality and computational costs. The practical value of the method lies in the possibility of its application in automated monitoring systems, video analytics and decision support, including military intelligence and cybersecurity tasks.

Keywords

Feature extraction, recurrent estimates, covariance, streaming data, classification, military objects[1]

1. Introduction

Feature extraction is closely connected with a classification problem is one of the fundamental and most important stages of building any intelligent system that works with data. Regardless of whether we are talking about medical diagnostics based on tests, automatic speech recognition, predicting customer creditworthiness, or classifying images, it is the stage of constructing the feature space that largely determines the ultimate success of the model [1-3]. No matter how powerful a classifier is, it always depends on how informative and relevant the object's characteristics were extracted and passed to the input. The most modern machine learning methods, such as deep neural networks, although they have the ability to independently form internal representations, essentially solve the same problem: they find such transformations of the original data that turn a set of signals into a feature space convenient for separating classes [4-6].

Classification is the task of finding a surface that separates the points of one class from the points of another. The shape and position of this boundary depend on how well the feature space itself is chosen. If the features poorly reflect the essence of the objects, the classes mix up and become indistinguishable. If the features are chosen well, the objects form separate clusters, and the separation task becomes much easier.

This idea is especially important in situations where there are massive of little data. Many modern machine learning algorithms require large samples for high-quality training. But in medicine, finance, or other areas, you have to work with a limited number of examples. Here, good

ProfIT AI'25: 5th International Workshop of IT-professionals on Artificial Intelligence, October 15–17, 2025, Liverpool, UK

[1*] Corresponding author.

volodymyr.kartashov@nure.ua (V. Kartashov); yevgeniy.bodyanskiy@nure.ua (Ye. Bodyanskiy); nonna.kulishova@nure.ua (N. Kulishova); yehor.kulishov@nure.ua (Ye. Kulishov)

0000-0001-8335-5373 (V. Kartashov); 0000-0001-5418-2143 (Ye. Bodyanskiy); 0000-0003-1142-4100 (N. Kulishova); 0009-0008-8731-8500 (Ye. Kulishov)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-02-07

CEUR-WS.org/Vol-4164/paper15.pdf

features can compensate for the lack of data. For example, a patient's medical indicators themselves may be disparate, but correctly constructed combinations of these indicators provide key information for making a diagnosis [7].

The history of computer vision clearly shows the importance of the method for forming features. Before the advent of neural networks, researchers used image characteristics such as gradients, textures, and shape moments to automatically generate a description of properties. These features ensured resistance to changes in illumination or scale and allowed algorithms to distinguish objects. Modern deep networks only automate this process, but their essence is the same - constructing a feature space where objects of different classes become distinguishable [4].

In text processing, the situation is similar: raw texts cannot be fed directly to the algorithm. Therefore, vector representations are created: from simple bags of words and TF-IDF to complex embeddings that reflect the meaning of words and their context [8, 9]. It was the emergence of various feature representations that became a breakthrough, allowing to significantly improve the quality of text classification.

In problems of processing audio information, it is necessary to use characteristics based on spectral analysis methods [10], which are extracted using, for example, convolutional and recurrent deep networks [11].

Most of the proposed methods for forming feature vectors in multimedia data streams are focused on the offline mode of the model, on the presence of significant data sets for training. Accordingly, it is assumed that in such problems it is possible to use complex architectures of models used for feature extraction, feature vectors can be quite large. However, in online streaming data processing problems, all this turns out to be unavailable or technically unrealizable. There is a need to create small-dimensional feature vectors using fast calculation algorithms. These requirements are met by statistical characteristics - average values, variance, covariance coefficient, which are calculated within a window sliding along the flow of constantly updated data.

It is this problem - the search for a compromise between the quality of the feature description and the efficiency of its calculation - that is actively addressed in modern literature. Thus, in [12], a hybrid model COCALITE is proposed that combines a compact architecture with a set of statistical features. This approach combines the advantages of a deep model capable of automatically extracting complex dependencies and a carefully selected set of interpretable features. The result is an increase in classification accuracy with a sharp reduction in the number of model parameters (only 4.7% of Inception), which is critically important in conditions of limited resources and when working with streaming data.

In cybersecurity tasks, where input data is high-dimensional and contains redundant information, the emphasis is on the systematic selection of the most significant characteristics. Logeswari and colleagues [13] proposed the Synergistic Dual-Layer Feature Selection (SDFC) algorithm, which combines statistical methods (mutual information, variance threshold) and model-oriented approaches (SVM with recursive feature elimination, PSO). Unlike COCALITE, which focuses on combining manual and deep features, here we are talking about a multi-stage reduction of the feature space before feeding data to the LightGBM and XGBoost classifiers. This combination made it possible to achieve high accuracy of attack detection in the IoT environment with lower computational costs. However, both hybrid architectures and two-level selection more often involve working in offline processing conditions, when it is possible to calculate features in advance and train complex models. In streaming scenarios with changing data, the ability to adaptively update the set of used features becomes key. This problem is devoted to the work [14], which proposes methods for online filtering of features for streaming data with conceptual drift. Their algorithms allow real-time revision of feature significance, while maintaining computational ease and without sacrificing accuracy. The authors have shown that online screening is capable of reproducing selection quality comparable to offline methods, while providing lower memory and time costs. Most importantly, the integration of model adaptation increases the probability of correctly identifying “truly significant” features in the context of changing data statistics.

If we consider these trends as a whole, we can identify a key trend: methods for constructing a feature space tend to combine expressiveness (due to deep architectures or extended feature sets), compactness (through strict selection and regularization), and adaptability (through online updating of feature significance). In streaming media classification problems, it is the latter characteristic that comes to the fore. This requires recurrent updating of statistics, such as mean values, variances, or covariances, which form compact but informative feature vectors.

2. Related works and problem statement

The problem of covariance estimation occupies a central place in modern statistics, machine learning, and engineering applications. The quality of signal filtering, the reliability of localization, the accuracy of forecasts, and the adequacy of statistical inference depend on the correctness of the covariance structure restoration. However, in real-world problems, researchers face a number of limitations: limited data volumes, high dimensionality of the feature space, the presence of noise and drift, as well as the need to work online when information is received continuously. These challenges have generated interest in recurrent (online) methods for estimating covariance matrices, which update estimates as data arrives, without requiring storage of the entire sample.

One of the illustrative examples of the application of such methods is localization systems in intelligent transport. Traditional odometry suffers from the accumulation of bias errors, which leads to incorrect uncertainty modeling. In [15], the Drift Covariance Estimation strategy was proposed, which allows refining the covariance of odometrical errors using readings from additional sensors that are not subject to drift. Recursive updating of the covariance matrix makes it possible to adapt the system to changes in external conditions and gradually reduces the uncertainty in localization models. The advantage of the approach is integrability into standard filters (EKF, UKF, H∞), which significantly increases their stability. However, the disadvantage remains the dependence on the presence of auxiliary sensors and the risk of incorrect accounting for errors if their statistical nature changes significantly. The theoretical basis of the algorithm is based on the approximation of drift using external observations, which makes the method applicable in real-world conditions, although strict optimality guarantees are not always feasible.

Another set of problems is related to modeling, where it is necessary to estimate covariances in spaces of huge dimensions with extremely limited samples. Vishny and colleagues [16] emphasized that classical statistical methods lose their validity in such a situation, since the number of observations is smaller than the problem dimension. To overcome this problem, they have proposed recurrent procedures in which covariances between variables are dynamically "discounted" depending on the noise level. This is actually a type of regularization built into the estimation process, which allows avoiding overfitting and maintaining stability. The advantage of the approach is that the algorithm has low computational complexity and can work in conditions of streaming data. The disadvantage is that the methods require knowledge or approximation of the noise level, which can be difficult in problems with a heterogeneous error structure. From a theoretical point of view, the authors ensure the preservation of key properties of the covariance matrix, which makes the method statistically correct and applicable for data assimilation in complex models. Significant progress has also been made in the field of stochastic optimization. Machine learning problems that use stochastic gradient descent methods require not only finding the optimal solution, but also the ability to estimate confidence intervals for the model parameters. Here, recurrent covariance estimation allows us to embed statistical inference directly into the learning process. Zhu et al. [17] have proposed an online estimator of the covariance matrix for averaged SGD iterates. The algorithm updates the estimate when new observations are received, without requiring storage of the entire iteration history. The advantages are obvious: efficiency in terms of memory and computational costs, the ability to construct asymptotically correct confidence intervals on the fly. Limitations are related to the sensitivity to the choice of the gradient descent step and the need to accumulate a sufficient number of iterations for the asymptotic properties to manifest. The theoretical justification of the method is based on classical results on the normality of averaged iterates, which guarantees the consistency and convergence of the proposed estimator. The development of this idea can be seen in a more recent paper [18], which considered much more complex problems of non-smooth and non-convex variational inclusions. Unlike smooth convex scenarios, where the theoretical analysis has long been worked out, the situation here is complicated by the lack of monotonicity and regularity. The authors proposed a recursive method based on batch means, which groups a sequence of iterates and estimates the covariance over these groups. This approach eliminates the need to know the sample size in advance and allows for online adaptation. An important advantage is that the method achieves a convergence rate comparable to the best known results in simpler scenarios, despite the complexity of the problem. A disadvantage is the need to carefully select the sequence of batch sizes, otherwise the efficiency drops sharply. From the theoretical point of view, the work is significant in that it was the first to provide strict guarantees of the consistency of covariance estimates in non-smooth and non-convex conditions, which opens the way to correct statistical inference even in very complex optimization problems.

Engineering applications also demonstrate the importance of recursive covariance estimation. Kalman filters and their variants are traditionally used in dynamic structure identification problems. However, the efficiency of these methods decreases sharply in the case of ill-conditioned systems caused by the sensor network architecture. Liu et al. [19] have proposed a new recursive smoothing method for estimating states and inputs of vibrating structures. Unlike existing minimum-variance unbiased smoothers, their method is applicable to both feedforward and rank-deficit systems. The key advantage is that the method does not require a priori information on the input statistics and adapts to observed data. The disadvantage is that the algorithm is essentially focused on linear systems, and the extension to nonlinear scenarios remains open. The theoretical basis of the method is related to a new discrete-time indexing, which allows to bypass the limitations of classical MVU approaches. The authors confirmed the validity and efficiency of the method using numerical examples, comparing it with several versions of the Kalman filter.

If we consider all these studies together, we can notice a number of common patterns. Recursive estimation of covariances is primarily motivated by the need to work under conditions of limited resources: limited data volume, limited memory, or limited computation time. In many cases, it is not just about approximating the covariance structure, but about constructing algorithms that ensure asymptotic normality of estimates and allow statistical inference. The advantages of recurrent methods are obvious: they are adaptive, allow you to respond to changes in data properties, and often have low computational complexity. The disadvantages include sensitivity to algorithm parameters (batch size, learning step, regularization structure) and dependence on a priori assumptions, which are not always met in real applications.

From the point of view of theoretical foundations, three levels can be distinguished. In applied problems, as in [15] and [19], the correctness of the methods is confirmed primarily experimentally and through the stability of filters. In high-dimensional and small-sample problems, as in [16], the proposed procedures are justified by preserving the structural properties of covariance matrices, which guarantees their use in modeling. Finally, in stochastic optimization, as in [17] and [18], the emphasis is on rigorous proofs of consistency and convergence rate, which allows embedding covariance estimation in mathematically sound statistical inference procedures. That is why this paper focuses on the formation of a feature space based on online covariance estimation — as a natural development of the ideas embedded in hybrid and adaptive methods in modern literature, and considers the problem of creating a feature vector based on recurrent online covariance estimation for the problem of classifying streaming multimedia data.

3. Materials and methods

Recurrent calculation of mean, variance and covariance for streaming data plays a key role in modern intelligent systems operating in real time. Unlike offline processing, where the entire array of information can be downloaded and analyzed in advance, in a streaming scenario, data arrives continuously and often in large volumes. It is impossible to store the entire stream either in memory or in processing time, so it is necessary to rely on recurrent formulas that allow updating statistics step by step. The formulas are derived based on the principle of optimal recursive Kalman estimation, which implements the process of parametric estimation based on the autoregressive model of the signal generation process (Fig. 1):

$$\bar{x}_k(k)=\bar{x}_k(k-1)+\frac{1}{k}(\bar{x}_k(k)-\bar{x}_k(k-1)), \tag{1}$$

where $x(\tau), \tau=1,2,\ldots,k$ is the sequence of input signals, k is the current discrete time.

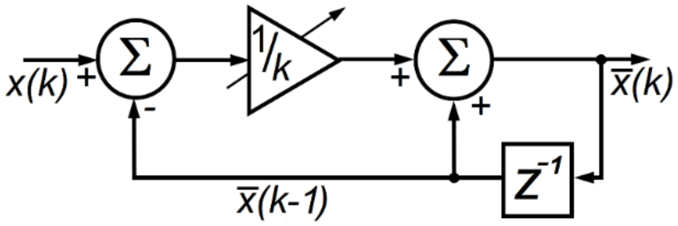

Figure 1: Recurrent model for calculating the average value of consecutive discrete data.

The average value calculated recurrently provides quick control over the central tendency of the flow, and allows for timely recording of shifts or changes in the signal level.

The variance updated in the flow reflects the degree of variability of the data and helps to identify areas with abnormally high or low variability (Fig. 2):

$$\sigma_x^2(k)=\sigma_x^2(k-1)+\frac{1}{k}((\bar{x}_k(k)-\bar{x}_k(k-1))^2-\sigma_x^2(k-1)), \tag{2}$$

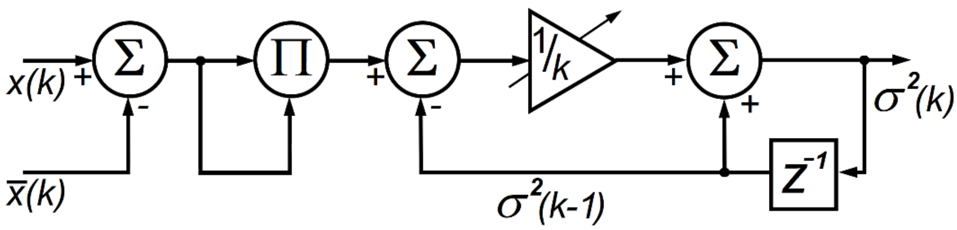

Figure 2: Recursive model for calculating the variance of sequential discrete data.

Covariance calculated recursively is especially important when it is necessary to track the connections between features: their appearance, disappearance or change in the manifestation of connections. However, in this paper it is proposed to calculate the covariance not between features, but between several adjacent points of sequential data (Fig. 3):

$$r_x(k,d)=r_x(k-1,d)+\frac{1}{k}((\bar{x}_k(k)-\bar{x}_k(k-1))(\bar{x}_k(k)-\bar{x}_k(k-1))-r_x(k-1,d)), \tag{3}$$

where $d=1,2,\ldots,p$ is the number of data points taken into account by the recurrent covariance.

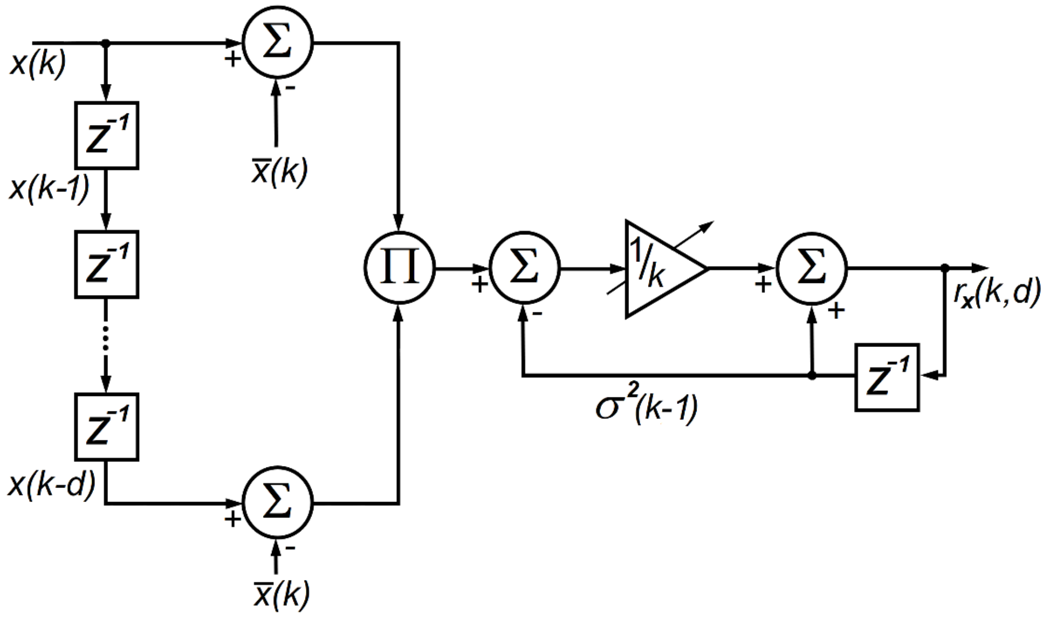

Figure 3: Recurrent model for calculating the covariance d of consecutive discrete data.

Each of the proposed recurrent models can be represented as a separate block, from which a module for forming a feature vector is formed (Fig. 4).

For the input sequence of discrete data, the module creates a feature vector

$$X(k)=(\bar{x}_k(k),\sigma_x^2(k),r_x(k,2),\ldots,r_x(k,d))^T, \quad (4)$$

of dimension (d+2)×1.

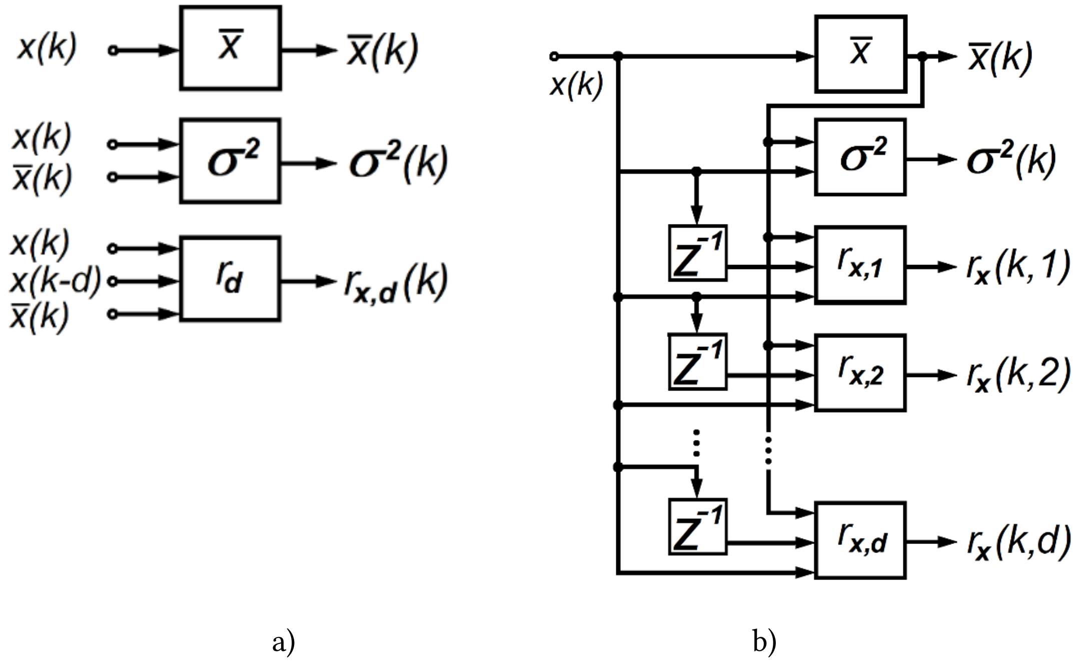

Figure 4: Architecture of the feature vector generation module using recurrent mean, variance and covariance estimation blocks; a) individual blocks; b) module architecture.

The advantage of such methods is that each new observation can be taken into account in constant time, without recalculating the entire history. This makes the algorithms computationally efficient and robust to large amounts of data. Recurrent statistics allow streaming systems to adapt to changes in input data, which improves classification accuracy, forecast reliability, and anomaly detection timeliness.

4. Main results

The effectiveness of the proposed approach to feature vector generation was experimentally evaluated on a military object recognition task. The dataset [20] comprises 3416 images of personnel and equipment across eight categories (artillery, infantry fighting vehicles, UAVs, armored vehicles, armored personnel carriers, infantry, multiple rocket launchers, and tanks), with varying viewpoints and conditions. Bounding box annotations enable object extraction and classification, though class distribution is imbalanced. Images contain either single or multiple objects from the same or different classes.

Device properties:

- processor: AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz;
- RAM: 16.0 GB;
- memory: 954 GB SSD WDC PC SN730 SDBPNTY-1T00-1101;
- video adapter: NVIDIA GeForce RTX 3060 Laptop GPU (6 GB);
- system type: 64-bit operating system, x64 processor.

Several different approaches were chosen for classification: Optimizable Tree; Weighted KNN; Optimizable KNN; Efficient Logistic Regression; Efficient Linear SVM; Optimizable Neural Network; Optimizable Naïve Bayes; Optimizable Ensemble; LVQ network. 5-fold cross validation was used for every model.

The data is split into training and test sets (0.7:0.3). Thus, the size of the training sample is 2392 images, the test sample is 1024 photos. Preprocessing included grayscale conversion and resizing to 80 × 60 pixels and subsequent vectorization. Two series of experiments were conducted. In the first series, the input feature vector included only image pixels $X(k)=x(\tau)^T, \tau=1,2,\ldots,k$. In the second series, the input feature vector was collected according to the proposed approach:

$$X(k)=(\bar{x}_k(k), \sigma_x^2(k), r_x(k,2), \ldots, r_x(k,d))^T, d=2 \quad (5)$$

The results of image classification of the dataset are presented in Table 1 and Table 2.

The optimized ensemble demonstrated the highest accuracy of 75.5% in solving the problem using the expanded feature vector. Its parameters are: Learner type: Decision Tree; Ensemble method: Bag; Number of splits: 2309; Number of learners: 476; Hyperparameter Search Range Ensemble method: Bag, Boost, RUSBoost; Number of learners: 10-500; Learning rate: 0.001-1; Optimizer: Bayesian optimization.

Table 1
Results of training models when solving the problem of classification of military objects by photos, when the input feature vector contains only image pixels

Model	Accuracy, %	Prediction speed, ob/sec	Training time, sec	Model size
Optimizable Tree	59.6	380	1436.5	743 kB
Weighted KNN	71.4	410	28.9	88 kB

Optimizable KNN	72.7	310	576.7	88 kB
Efficient Logistic Regression	20.2	820	30.4	20 MB
Efficient Linear SVM	27.4	810	37.7	20 MB
Optimizable Neural Network	70.7	790	14052	5 MB
Optimizable Naïve Bayes	63.6	6.6	31794	432 MB
Optimizable Ensemble	**75.7**	410	27955	359 MB

Table 2
Results of training models when solving the classification problem based on photographs of military objects with an extended vector of input features, including mean, variance and covariance

Model	Accuracy, %	Prediction speed, ob/sec	Training time, sec	Model size
Optimizable Tree	60,2	68	5281,9	3 MB
Weighted KNN	69,3	37	303	353 MB
Optimizable KNN	72.4	8,7	8790	353 MB
Efficient Logistic Regression	21,5	52	268,3	81 MB
Efficient Linear SVM	28.4	52	292	81 MB
Optimizable Neural Network	73,3	54	1,28e+05	47 MB
Optimizable Naïve Bayes	63.6	2	1,15e+05	2 GB
Optimizable Ensemble	**75.5**	46	1,63e+05	1 GB

The AUC values for some classifiers are given in Table 3.

Table 3
AUC by classes for some models when solving the classification problem based on photographs of military objects with an extended vector of input features, including mean, variance and covariance

Model	AUC by classes							
	1	2	3	4	5	6	7	8
Optimizable Tree	0.8421	0.8204	0.768	0.7983	0.809	0.7905	0.8039	0.8608
Weighted KNN	0.9473	0.9464	0.9205	0.9233	0.9246	0.9291	0.9422	0.9692

Optimizable KNN	0.9607	0.9568	0.9344	0.9374	0.9419	0.9395	0.9406	0.9595
Optimizable Neural Network	0.9494	0.9302	0.9177	0.9278	0.9283	0.9204	0.9281	0.9548
Optimizable Naïve Bayes	0.8944	0.8955	0.8981	0.8876	0.8989	0.8991	0.8957	0.9062
Optimizable Ensemble	0.9645	0.9604	0.9435	0.9513	0.9539	0.949	0.9443	0.9667

Confusion matrix, ROC and Minimum classification error plot for the optimized ensemble are shown in Figures 5-7.

The experiments confirmed the effectiveness of the proposed approach to forming a feature vector for the task of classifying military images. Comparison of two series of experiments showed that using an extended vector, including statistical characteristics and autocovariance features, provides higher recognition quality compared to the option where the vector was formed exclusively from pixel values. This indicates that additional information about the image structure and the relationships between elements allows classification algorithms to more effectively separate objects into classes.

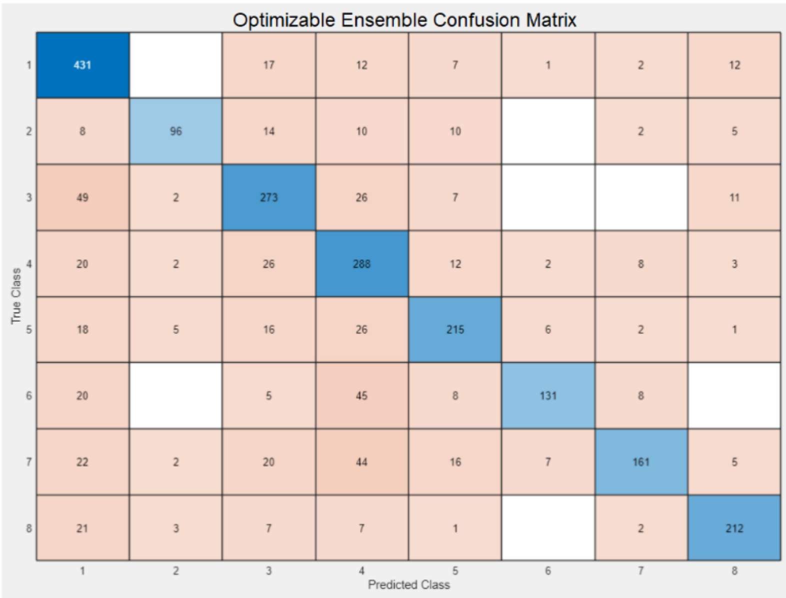

Figure 5: Confusion matrix for the optimized ensemble when solving the classification problem based on photographs of military objects with an extended vector of input features, including mean, variance, and covariance.

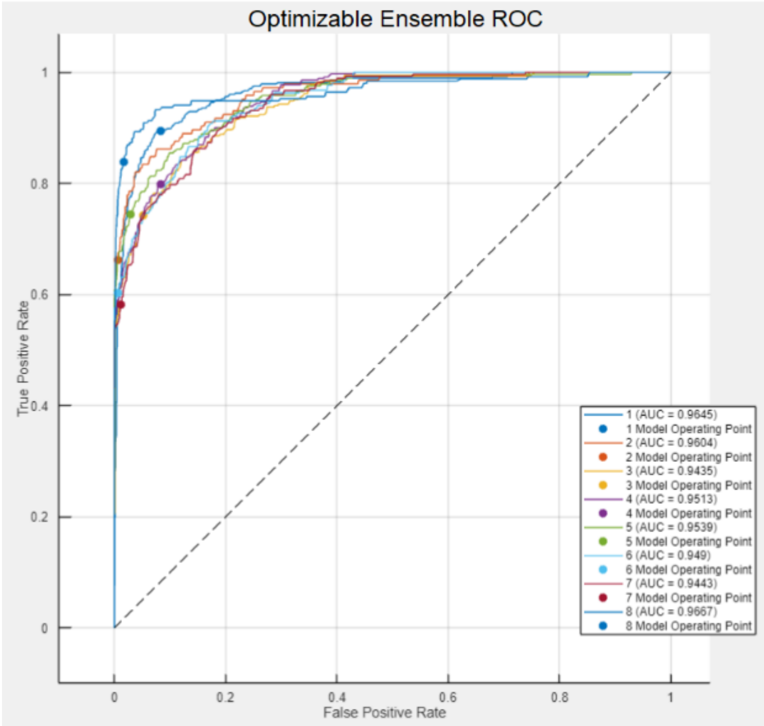

Figure 6: ROC for the optimized ensemble when solving the classification problem of military objects from photographs with an extended vector of input features, including mean, variance and covariance.

The results of the experiments show that the proposed approach to forming a feature vector has practical value for the tasks of automatic recognition of military objects. Using an extended vector, including statistical and covariance characteristics of images, made it possible to significantly increase the accuracy of classification compared to simply taking into account the brightness values of pixels. This opens up opportunities for developing more reliable systems for analyzing visual data in conditions of limited image quality, different shooting angles, and a complex background.

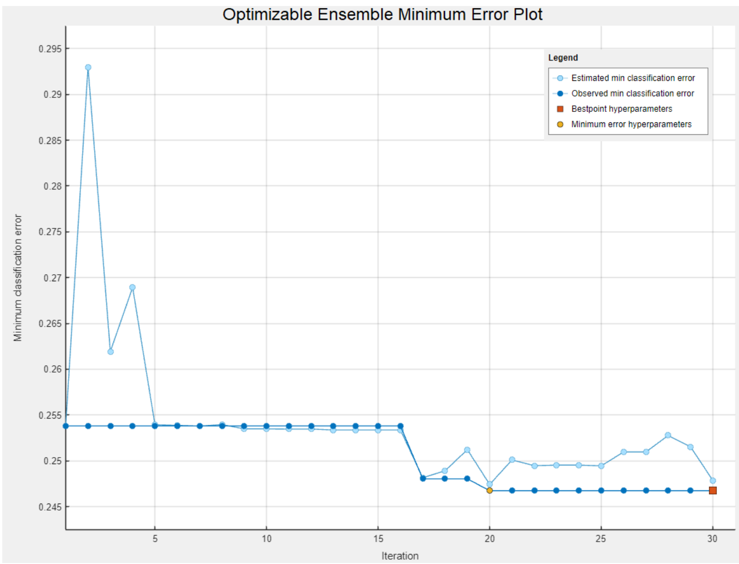

Figure 7: Minimum classification error plot for the optimized ensemble when solving the classification problem based on photographs of military objects with an extended vector of input features, including mean, variance, and covariance.

5. Conclusions

The paper considers the issue of forming a feature vector for solving classification problems based on streaming data, for which it is proposed to expand the vector by including recurrently estimated mean value, variance, and covariance. The calculation relationships are given and the architecture of the streaming data preprocessing module is proposed, which forms an extended feature vector using recurrent estimates.

Two series of experiments were conducted, during which the preprocessing module was used to solve the problem of classifying images of military objects.

A comparison of the results of the two series of experiments shows that the choice of the classification method and the formation of the feature vector have a significant impact on both the recognition accuracy and the computational characteristics of the models.

Firstly, one can note a general improvement in the quality of classification when moving from the first series to the second. For most algorithms, an increase in accuracy of 1–3% is observed, most noticeable for the optimized neural network (from 70.7% to 73.3%) and the optimized decision tree (from 59.6% to 60.2%). The optimized ensemble demonstrated the highest quality in both series, providing 75.7% and 75.5%, respectively. This confirms that ensemble methods remain the most effective for multi-class classification problems in the presence of data heterogeneity.

Secondly, the improvement in accuracy is accompanied by an increase in resource requirements. In the second series, the models became noticeably “heavier”: the size of the optimized ensemble increased from 359 MB to 1 GB, and the neural network — from 5 MB to 47 MB. The training time also increased significantly: for the ensemble — from ~28 thousand seconds to more than 160 thousand seconds, for the neural network — from 14 thousand to 128 thousand seconds. This indicates that the inclusion of advanced features increases the load on the computing infrastructure.

Thirdly, the prediction speed decreased: for example, for KNN and the ensemble, the drop was almost an order of magnitude. This makes such models less suitable for real-time tasks. Thus, the use of an extended feature vector improves the quality of recognition, but requires a compromise between accuracy, resources, and prediction speed. For practical application in online systems, compromise models (for example, KNN or decision trees) are preferable, while ensembles and neural networks are advisable to use in offline analytics.

Thus, it can be concluded that the proposed feature generation method is a promising direction for object recognition tasks in complex conditions. Its application allows increasing the efficiency of both traditional machine learning algorithms and ensemble models.

The practical significance of this approach lies in the possibility of its implementation in automated surveillance, monitoring, and decision support systems. Automatic classification of objects, such as enemy equipment or manpower, can improve reconnaissance efficiency and reduce the workload of operators. In the future, the method can be integrated into onboard systems of unmanned aerial vehicles, video analytics, or security systems, ensuring timely and accurate target identification.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 in order to Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

[1] C. Xueyi. A Comprehensive Study of Feature Selection Techniques in Machine Learning Models. Insights in Computer, Signals and Systems 1(1) (2024). DOI: http://dx.doi.org/10.2139/ssrn.5154947.

[2] M. V. R. Sarobin, J. Ranjith, D. Ashwath, K. Vinithi, Smiti, V. Khushi. Comparative Analysis of Various Feature Extraction Methods on IoT 2023. Procedia Computer Science 233 (2024) 670–681. DOI: https://doi.org/10.1016/j.procs.2024.03.256.
[3] Z. Fadhel, H. Attia, Y. Hussain. A Comparative Analysis of Feature Extraction Techniques for Fake Reviews Detection. Fusion: Practice and Applications (2025) 161–172. DOI: https://doi.org/10.54216/FPA.170212.
[4] W. Chao. Research on Features Extraction and Classification for Images based on Transformer Learning. Proceedings of 2024 International Conference on Machine Learning and Intelligent Computing. Proceedings of Machine Learning Research 245 (2024): 67–75. URL: https://proceedings.mlr.press/v245/chao24a.html.
[5] Z. Li, H. Wang, X. Jiang. AudioFormer: Audio Transformer learns audio feature representations from discrete acoustic codes. arXiv:2308.07221v5 [cs.SD], 23 Aug (2023).
[6] Ye. V. Bodyanskiy, N. Ye. Kulishova, V. P. Tkachenko. Feature vector generation for the facial expression recognition using neo-fuzzy system. Radio Electronics, Computer Science, Control 3 (2018). DOI: https://doi.org/10.15588/1607-3274-2018-3-10.
[7] I. A. Mageed. Entropy-based feature selection with applications to industrial internet of things (IoT) and breast cancer prediction. Big Data and Computing Visions 4(3) (2024)170–179. DOI: 10.22105/bdcv.2024.479315.1203.
[8] S. Singh, K. Kumar, B. Kumar. Analysis of feature extraction techniques for sentiment analysis of tweets. Turkish Journal of Engineering 8(4) (2024) 741–753.
[9] A. Muqadas, H. U. Khan, M. Ramzan, et al. Deep learning and sentence embeddings for detection of clickbait news from online content. Scientific Reports 15 (2025) 13251. DOI: https://doi.org/10.1038/s41598-025-97576-1.
[10] M. de Brito Santos, R. de Moraes Calazan. Improved spectral dynamic features extracted from audio data for classification of marine vessels. Intelligent Maritime Technology Systems 2 (2024) 18. DOI: https://doi.org/10.1007/s44295-024-00029-0.
[11] A. Yousuf, D. S. George. Feature extraction of audio data for speaker's gender classification. Journal of Physics: Conference Series 2998(1) (2025) Art. 012003. DOI: 10.1088/1742-6596/2998/1/012003.
[12] O. Badi, M. Devanne, A. Ismail-Fawaz, J. Abdullayev, V. Lemaire, S. Berretti, J. Weber, G. Forestier. COCALITE: A Hybrid Model Combining Catch22 and LITE for Time Series Classification. 2024 IEEE International Conference on Big Data (BigData) (2024) 1229–1236. DOI: 10.1109/BigData62323.2024.10825872.
[13] G. Logeswari, K. Thangaramya, M. Selvi, et al. An improved synergistic dual-layer feature selection algorithm with two type classifier for efficient intrusion detection in IoT environment. Scientific Reports 15 (2025) 8050. DOI: https://doi.org/10.1038/s41598-025-91663-z.
[14] M. Wang, A. Barbu. Online Feature Screening for Data Streams With Concept Drift. IEEE Transactions on Knowledge and Data Engineering 35(11) (2023) 11693–11707. DOI: https://doi.org/10.1109/TKDE.2022.3232752.
[15] M. Osman, A. Hussein, A. Al-Kaff, F. García, D. Cao. A Novel Online Approach for Drift Covariance Estimation of Odometries Used in Intelligent Vehicle Localization. Sensors 19(23) (2019) 5178. DOI: https://doi.org/10.3390/s19235178.
[16] D. Vishny, M. Morzfeld, K. Gwirtz, E. Bach, O. R. A. Dunbar, D. Hodyss. High-dimensional covariance estimation from a small number of samples. Journal of Advances in Modeling Earth Systems 16 (2024) e2024MS004417. DOI: https://doi.org/10.1029/2024MS004417.
[17] W. Zhu, X. Chen, W. B. Wu. Online Covariance Matrix Estimation in Stochastic Gradient Descent. arXiv:2002.03979v3 [stat.ML], 22 Jun (2021).
[18] L. Jiang, A. Roy, K. Balasubramanian, D. Davis, D. Drusvyatskiy, S. Na. Online Covariance Estimation in Nonsmooth Stochastic Approximation. arXiv:2502.05305v1 [stat.ML], 7 Feb (2025).

[19] Z. Liu, M. E. Hassanabadi, D. Dias-da-Costa. A linear recursive smoothing method for input and state estimation of vibrating structures. Mechanical Systems and Signal Processing 222 (2025) 111685. DOI: https://doi.org/10.1016/j.ymssp.2024.111685.

[20] GonTech. War_tech_v2.0: Detection objects dataset. Kaggle (2024). URL: https://www.kaggle.com/datasets/gon213/war-tech-v2-0-by-gontech.