



UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea Magistrale in Informatica

Tesi di Laurea

Rilevamento di anomalie nelle infrastrutture della rete elettrica: anomalie
del consumo energetico mediante apprendimento automatico

Detecting anomalies in power grid infrastructures: power consumption
anomalies using machine learning

Madina Kuldeeva

Relatore/Relatrice: Tomasso Zoppi

Anno Accademico 2024-2025

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text. Translation, adaptation in whole or in part, reproduction using any means (including media, microfilm and photocopy formats) and electronic storage are reserved in all countries.

Signature

A handwritten signature in black ink, appearing to be 'J. J. J.' or similar, written in a cursive style.

Detecting anomalies in power grid infrastructures: power consumption anomalies using machine learning

Madina Kuldeeva

Abstract

In recent years, intelligent power systems (Smart Grids) have played a crucial role in ensuring effective monitoring and management of power consumption. However, the growth of smart counters makes such systems vulnerable to false data injection (FDE), which can affect the accuracy of forecasts, pricing and network balancing.

The aim of this work is to develop a methodology for detecting anomalies in smart counters data caused by false data introduction, using machine learning methods. The study includes an analysis of existing approaches to anomaly detection, artificial injection of various types of anomalies (noisy, skipping, zero values and others) and application of modern algorithms such as random forests and gradient boosting. In the course of work it is proposed to use data sets on power consumption of households, urban facilities and industrial buildings, applying metrics Precision, Recall, F1-score for evaluating the quality of models.

The scientific novelty of research is in adaptation and application of modern approaches of machine learning to the identification of specific threats in intelligent power systems. The practical importance of the work is the possibility to integrate the developed methodology into energy network monitoring systems, which will allow to increase their reliability and resistance to threats.

The main results include qualitative visualization of detected anomalies, analysis of importance of features and recommendations for further use of developed methods to ensure safety and reliability of intelligent power systems.

Keywords

Power consumption, power grid, anomalies, machine learning, smart grid, smart meter

Detecting anomalies in power grid infrastructures: power consumption anomalies using machine learning

Madina Kuldeeva

CHAPTER 1. INTRODUCTION	6
CHAPTER 2. REVIEW OF LITERATURE AND TECHNOLOGY	8
2.1 History and development of energy systems: from centralized networks to smart technologies	8
2.2 Smart Grids: state, benefits, main security threats.....	9
2.3 Smart meters: architecture, data exchange protocols, vulnerabilities.	16
2.4 Methods of anomaly detection: statistical approaches, machine learning, deep learning, hybrid systems.	20
2.4.1 Statistical approaches to anomaly detection.....	23
2.4.2. Machine-learning techniques of anomaly detection.....	24
2.4.2.1 Decision Trees & Random Forests:.....	24
2.4.2.2 Gradient Boosting (XGBoost, LightGBM, CatBoost)	24
2.4.2.3 Clustering (e.g., k-medians, DBSCAN, Gaussian Mixture Models).....	25
2.4.2.4 Regression models (linear and polynomial regression, neural network approaches)	26
2.4.3 Deep learning methods for anomaly detection	26
2.4.3.1 Recursive neural networks (RNN) and LSTM.....	27
2.4.3.2 Autoencoders.....	27
2.4.3.3 Transformer-based architecture (e.g., BERT, Time Series Transformer) ..	28
2.4.4 Hybrid systems for anomaly detection	28
2.4.4.1 Combination of statistical methods and machine learning	28
2.4.4.2 Use of deep learning models to extract features and machine learning for classification	29
2.4.4.3 Combining several models (ensemble methods)	29
CHAPTER 3. METHODOLOGY OF ANOMALY DETECTION	31
3.1 Methodological approach	31
3.2 Source data sets: where the data, data structure, and timescale (hourly, weekly) came from.....	32

3.3 Format and characteristics of the source data: volume, number of features, time series.....	33
3.4 False data injection process and types of anomaly	37
3.5 Model and Algorithm Selection: Decision Tree, Random Forest, Gradient Boosting, and Neural Networks.....	42
3.6 Data Pre-processing and Feature Engineering	45
3.7 Metric selection: Precision, Recall, F1-score, Confusion Matrix, Accuracy evaluation.....	54
3.8 Research questions.....	60
3.9 Applied tools and libraries: python, scikit-learn, pandas, numpy, matplotlib, seaborn	61
CHAPTER 4. RESULTS OF EXPERIMENT	66
4.1 Quantitative results: analysis of model quality metrics	66
4.1.1 Experiment 1 - Impact of the delta feature.....	67
4.1.2 Experiment 2 - Use of lag-1 as an additional feature.	69
4.1.3 Experiment 3 - Smoothing the data with a moving average.	70
4.1.4 Experiment 4 - Combining delta and moving average.....	72
4.1.5 Experiment 5 - Use of the delta feature in ensemble models.....	74
4.1.6 Experiment 6 - Combined Features in XGBoost and Gradient Boosting.	75
4.2 Qualitative analysis: visualization of anomalies and analysis of the importance of features.....	77
4.3 Comparison with other studies	78
CHAPTER 5. DISCUSSION OF RESULTS	81
5.1 Interpretation of the data obtained and their comparison with theoretical expectations.	81
5.2 Strengths and weaknesses of the applied methods in the context of specific anomalies.....	84
5.3 Limitations of the study: applicability to real systems, quality of original data, possibility of scaling.	88
CHAPTER 6. CONCLUSION.....	90
RINGRAZIAMENTI.....	92
REFERENCES	93

CHAPTER 1. INTRODUCTION

Modern intelligent power systems, often referred to as Smart Grids, represent an evolution of traditional electric networks, allowing for more efficient energy management through the use of digital technologies and analytical tools. One of the most important elements of such systems is smart meters that collect data on power consumption in real time. These data become the basis for management decisions, load forecasting and optimization of energy supply processes.(Gungor, V. C., Sahin, D., Kocak, T., Ergüt, S., Buccella, C., Cecati, C., & Hancke, G. P., 2011)

Smart meters are an important element of such systems, being installed at the consumers' premises and providing detailed data on power consumption in real time. However, there are many examples of device tampering or hijacking, triggering incorrect operation of forecasting, pricing and balancing systems. There is therefore a growing need for automatic detection of anomalies due to these malicious events in power consumption time series.(Fang, X., Misra, S., Xue, G., & Yang, D., 2012)

However, the rapid development of smart grids has a number of challenges. One of the most acute problems is the injection of false data (FDI), which can disrupt the entire power grid infrastructure. Such attacks can lead to misforecasts, price distortions and power supply imbalances. The problem of false data is becoming especially relevant in the context of increasing integration of renewable energy sources and the increase of the amount of information processed.(Xie, L., Chen, Y., & Kumar, P. R., 2010)

Using machine learning to detect anomalies in data is a promising direction. Machine learning techniques allow to identify complex and hidden patterns in data, making them crucial tools for detecting malicious activities aiming at disrupting of energy systems. The development and implementation of such algorithms allows to minimize risks related with the injection of false data, and ensure the stable action of power supply systems.(Amiri-Zarandi, M., Dara, R., Dara, R., Duncan, E., & Fraser, E., 2022)

Within this work, the potential of modern machine learning methods to solve the problem of false data detection will be considered, and recommendations for their implementation in real operation will be presented.

The scientific objective of this work is to develop and evaluate a methodology that allows to effectively detect anomalies in power consumption data using machine learning

methods. Several tasks are required to achieve the goal. First of all, it is necessary to examine the existing approaches to detect anomalies in power systems, highlight their main advantages and limitations. Second, consumption data is typically logged during normal operation, as companies are reluctant to share their data during ongoing attacks. As a result, an important part of the study will be the description of the process of artificial injection of anomalies into data, allowing to form training sets for supervised machine learning models.

The next step will be to apply and compare different models of machine learning, including decision trees, random forests and other algorithms, to detect false data.

Pre-processing steps, including feature selection, cross-validation and model quality assessment will also be implemented. It should be noted that the data used are time-based, which requires special treatment in the application of machine learning methods. Some algorithms are initially oriented to work with time series, while others may need to create additional features that reflect time dependencies such as time lags, moving averages and other aggregates. The scientific novelty of research is the application of modern machine learning technologies to solve a specific problem of detecting false data in intelligent energy systems.

The practical importance of the work is that the proposed approaches can be implemented in energy network monitoring systems, which will allow to increase their reliability and resistance to cyber attacks and technical failures. Thus, this study contributes to the development of reliable and safe intelligent power systems capable of meeting modern challenges.

CHAPTER 2. REVIEW OF LITERATURE AND TECHNOLOGY

This chapter reviews literature and technology relevant to the thesis, providing a comprehensive background on smart power grids. Section 2.1 examines the historical development of power systems from centralized networks to contemporary Smart Grid technologies. Section 2.2 discusses Smart Grids in detail, highlighting their components, benefits, and critical cybersecurity threats. Section 2.3 covers smart meter technologies, including their architectures, communication protocols, and associated vulnerabilities. Section 2.4 explores anomaly detection methodologies, reviewing statistical approaches, machine learning techniques, deep learning, and hybrid systems.

2.1 History and development of energy systems: from centralized networks to smart technologies

The appearance of the first power grid in 1886 was an important step in the development of technology and society. The first network was centralized and intended for unidirectional transmission and distribution of electricity. By the beginning of the 20th century, a lot of local networks had appeared, which caused the need to merge them. This has made it possible to improve the reliability of the energy supply and reduce the costs of its maintenance.(Naamane, A., & M'Sirdi, K., 2013)

In the middle of the 20th century, power grids began to develop actively, especially in developed countries. Power plants were built near raw material sources, railways and ports, making them strategically important. When choosing places for hydroelectric power plants, the location of rivers was taken into account, and nuclear power stations were built near water bodies to cool equipment. The most polluting fossil-fuelled power plants have been trying to be located away from residential areas.

By the late 1960s, electricity was available in almost all developed countries except the most remote regions. That each consumer pays only for the actual electricity used, start to implement energy consumption accounting. There were also fixed-fee tariffs and double tariffs, which reduced the cost of electricity at night.

From the 1970s to the 1990s, demand for electricity increased significantly, requiring the construction of new power stations. However, not all regions were prepared for such changes, which led to deterioration of energy quality, accidents and voltage fluctuations. This has particularly affected industrial enterprises, heating systems and other organizations that have become increasingly dependent on stable electricity supplies.

At the end of the 20th century, an analysis of the intensity of demand for electricity was carried out. Peak loads in residential houses were caused by heating and cooling. For their smoothing began to use «peak generators», which worked short periods, but effectively covered high loads. These generators were inexpensive and could be started quickly.

In the 21st century, the integration of smart networks began, with China, India and Brazil being the pioneers. The term 'Smart Grid' was first used in 2003 to describe the capabilities of such systems. The main goals of smart grid implementation are related to the use of renewable energy sources, the transition from centralized systems to distributed and environmental improvement.(Bosnia and Herzegovina, 2023)

Smart grids help to improve the security of energy supply and manage distributed substations effectively. Automation can improve the monitoring and management of systems, making them more cost-effective and reliable.(www.elektro-expo.ru¹)

2.2 Smart Grids: state, benefits, main security threats.

Smart Grids are the evolution of traditional electric networks that use modern digital technologies, communication and measurement to improve efficiency, reliability and sustainability of energy supply.

Not all smart grids are equally developed. Many countries are actively working to upgrade obsolete distribution grids and turn them into intelligent systems. However, this process is complex and can take years, sometimes decades.(www.arrow.com²)

The performance of smart grid becomes clear if we consider its main components and key principles:

1. Advanced Metering Infrastructure (AMI):

- Unlike traditional grids, smart grids are characterized by a two-way exchange of information between producers, suppliers and end users, which allows to optimize energy flows in real time. The use of smart meters, Power quality monitoring systems and predictive algorithms contribute to improved energy efficiency and reduced operating costs. (www.injoit.ru³)

2. Grid Automation and Control:

¹ <https://www.elektro-expo.ru/ru/articles/smart-grid/>

² <https://www.arrow.com/en/research-and-events/articles/what-is-a-smart-grid-and-how-does-it-work>

³ <https://www.injoit.ru/index.php/j1/article/viewFile/1854/1728>

- The introduction of smart grid technologies has opened up possibilities for real-time monitoring and management of the power system. This helps to quickly detect problems such as power supply outages or voltage fluctuations and quickly eliminate them, which increases the reliability and efficiency of the system. More intelligent solutions allow the energy system to better meet the needs of modern society, while reducing costs and increasing its sustainability.

3. Renewable energy deployment:

- Smart grids use modern technologies such as sensors and communication systems to obtain real-time data on energy production and consumption. This helps to predict and manage distributed energy resources more accurately, simplifying the use of renewable energy sources in the power system. Such networks also support demand management programs that motivate consumers to reduce or increase power consumption depending on its availability. The development of smart grids has made renewable energy sources more efficient, increasing the reliability of energy systems and contributing to a more sustainable future. (www.grandslipring.com⁴)

4. Energy storage:

- Energy storage technologies, such as batteries, have become an important part of smart grids. They allow to save excess energy during periods of low demand and use it at peak hours. This helps to balance supply and demand, and facilitates the use of renewable energy in the grid. (ngurart.com.au⁵)

5. Demand management programs:

- Smart grids authorise consumers through demand management programmes. During peak demand or high electricity prices, consumers can voluntarily reduce their consumption or shift it to more relaxed periods. This helps to optimize the load on the network and maintain its stability. (bitech.tech⁶)

The realization of smart grid brings many benefits:

- They improve energy efficiency by optimizing the distribution of energy and reducing losses during transmission.

⁴ <https://www.grandslipring.com/hybrid-slip-rings/>

⁵ <https://ngurart.com.au/why-should-we-start-using-solar-power/>

⁶ <https://bitech.tech/knowledge-hub/120231228/smart-power-grid-vs-traditional-power-grid-part-2>

- Due to automation and extended monitoring systems become more reliable.
- Smart grids also promote the integration of renewable energy sources, reducing dependence on fossil fuels and reducing greenhouse gas emissions. (www.tdworld.com⁷)
- Real-time power consumption data allows users to make more informed decisions, reduce energy consumption and save on electricity costs.
- Furthermore, intellectual networks are opening up new opportunities for services, technologies and job creation in the energy sector. (www.prysmian.com⁸)

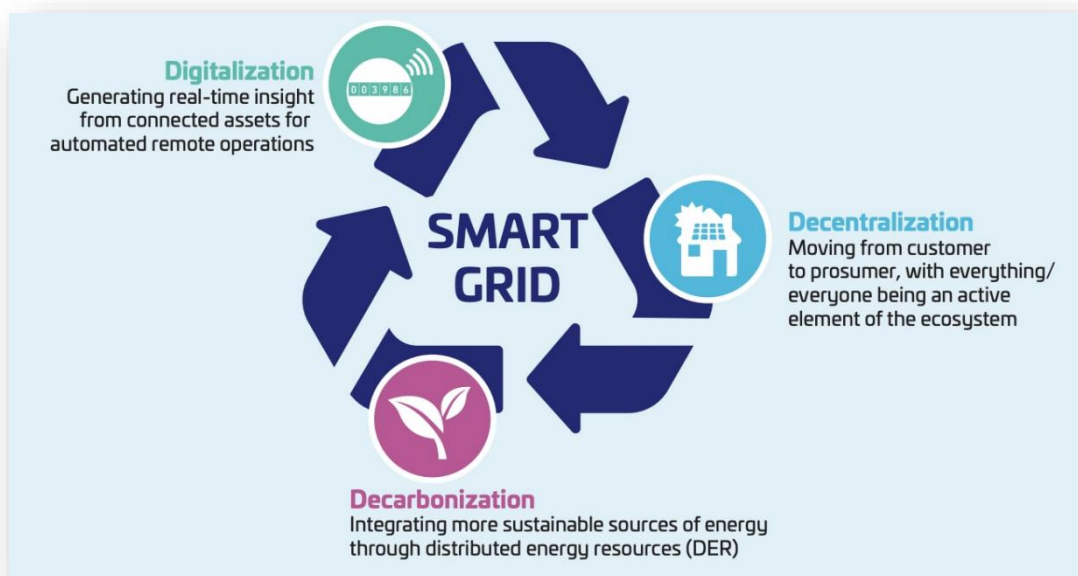


Figure 2.1: Smart grid advantages. Source: www.thalesgroup.com⁹

As we see in Figure 2.1 and as previously described, actually the smart grid has good potential, implementation and advantages.

However, despite the obvious advantages, intelligent power systems are exposed to a number of cyber threats and vulnerabilities.

Main areas where cybersecurity problems are most common:

⁷ <https://www.tdworld.com/overhead-transmission/article/20956550/abb-successfully-tests-ultrahigh-voltage-transformer-for-china-hvdc-transmission-link>

⁸ <https://www.prysmian.com/en/insight/sustainability/what-is-a-smart-grid-and-how-does-it-work>

⁹ <https://www.thalesgroup.com/sites/default/files/database/assets/images/2021-01/what%20is%20smart%20grid.jpg>

- **Confidentiality:** Installing smart meters in private homes allows utilities to collect data on habits and behavior of residents. This information may be of interest to different parties. For example, criminals can use electricity consumption data to determine when a house is empty and plan theft or other illegal activities.
- **Connection vulnerability:** As the connectivity capabilities of devices in smart grids increase, so does the risk of large-scale attacks. Intruders can access multiple components of the system because they are interconnected. If the hacker fails several network nodes, it can lead to serious failures, reduced efficiency and system corruption.
- **Security Management:** In smart grids, the number of devices that utilities need to monitor and manage is much higher. To maintain consumer trust, companies should ensure data security by monitoring possible cyber-attacks on individual devices. Such monitoring requires additional human resources, resources and computing power that were not required in traditional networks. (blockchain.ieee.org¹⁰)

Modern SCADA¹¹ systems use automation of distribution instead of manual labor to control the processes of distribution of electricity. Devices such as digital sensors and switches automate operations, monitor voltage, monitor equipment status, and control the power system as a whole. (blogs.cisco.com¹²)

The integration of SCADA into a network is crucial for improving the intelligent security and efficiency of the energy system. This security technology can provide significant benefits to smart grids, including protecting the privacy of citizens' data and minimizing service disruptions. (csiacy.org¹³)

Despite the existing cybersecurity problems, solutions can be found that will help minimize risks and take full advantage of smart grids. One such approach is to strengthen the protection of all potential entry points, which increases with automation. (blockchain.ieee.org¹⁴)

¹⁰ <https://blockchain.ieee.org/verticals/transactive-energy/topics/smart-grid-security-issues-cybersecurity-solutions-to-consider>

¹¹ Supervisory Control and Data Acquisition are designed for the management, monitoring and analysis of industrial devices and processes.

¹² <https://blogs.cisco.com/security/safety-first-business-second-security-none>

¹³ <https://csiacy.org/articles/the-efficacy-and-challenges-of-scada-and-smart-grid-integration/4/>

¹⁴ <https://blockchain.ieee.org/technicalbriefs/december-2018/blink-trusted-digital-identity-solution>

Consider some possible solutions:

- **Encryption:** It makes data unreadable without a key. This method allows to store data in encrypted form, which prevents their leakage even if intercepted.
- **Authentication:** Authentication and access control policies asks users to confirm their identity before accessing the system. Methods such as biometric authentication, use of certificates and tokens help to ensure that only authorized users can access.
- **Network security:** Virtual private networks (VPNs) protect data when it is transmitted over a network using encryption and authentication. This prevents access to data by other users.
- **Network intrusion detection and prevention systems:** These systems monitor network activity, identify and respond to potential threats by closing vulnerable entry points or updating security settings. (blockchain.ieee.org¹⁵)

These measures can be an effective complement to solutions aimed at addressing the cybersecurity problems of smart grids. They will significantly reduce the risk of attacks, provide data protection and maintain the reliability of the power system.

Well still, there is a problem of the introducing of false data (False Data Injection), which can distort the results of forecasting, load balancing or dynamic pricing systems. (www.injoit.ru¹⁶)

The False Data Injection (FDI) attack is one of the most dangerous cyber attacks. It is the deliberate introduction of distorted data into smart meter measurements, which compromises the accuracy of power system state estimation(SE). (www.sciencedirect.com¹⁷). Such attacks include swapping or changing data in the network management system, which makes measurements inaccurate. This could lead to wrong decisions and, in the worst case, serious disruptions to the power grid.

An attack with the introduction of false data can disrupt the normal operation of smart grids, distort data and even lead to large-scale power supply disruptions. This weakens the reliability and integrity of the system, affecting not only energy companies but also

¹⁵ <https://blockchain.ieee.org/verticals/transactive-energy/topics/smart-grid-security-issues-cybersecurity-solutions-to-consider>

¹⁶ <https://www.injoit.ru/index.php/j1/article/viewFile/1854/1728>

¹⁷ [https://www.sciencedirect.com/science/article/pii/S1364032122003306#:~:text=False%20Data%20Injection%20\(FDI\)%20%5B,bad%20data%20into%20meter%20measurements](https://www.sciencedirect.com/science/article/pii/S1364032122003306#:~:text=False%20Data%20Injection%20(FDI)%20%5B,bad%20data%20into%20meter%20measurements)

millions of consumers who depend on stable electricity for their daily needs. The consequences of such attacks are tangible: they can cause serious damage to the economy, security and well-being of the country. That is why it is important to confront such threats before the attackers become even more sophisticated and persistent. A successful attack on smart grids can have devastating consequences for the entire infrastructure. (Taher, M.A.; Behnamfar, M.; Sarwat, A.; Tariq, M., 2024)

A successful attack on the infrastructure of smart grid can have a devastating impact on the nation's well-being, economy, health and security. The attack with false data is one of the most dangerous. An example of such a threat was the 2015 attack when the Black Energy virus, penetrating through targeted phishing letters, gained access to the management systems of several regional electricity distribution companies. This virus was feeding false data, manipulating sensor readings and hiding the real state of the power grid. (Assante, M.J., july 2019)

The attack resulted in power outages affecting more than 200,000 customers in Ukraine. This incident is also connected with a series of other cyber attacks that occurred in Ukraine. For example, in 2016 a new attack led to a power outage, as a result of which Kiev lost about one fifth of its total electricity consumption during this period of the night. The national energy company "Ukrenergo" reported about this.(BBC, july 2019)

Figure 2.2 shows an example of a FDIA¹⁸ that affects sensor values. This case illustrates how such attacks can falsify system data.

¹⁸ False data injection attack

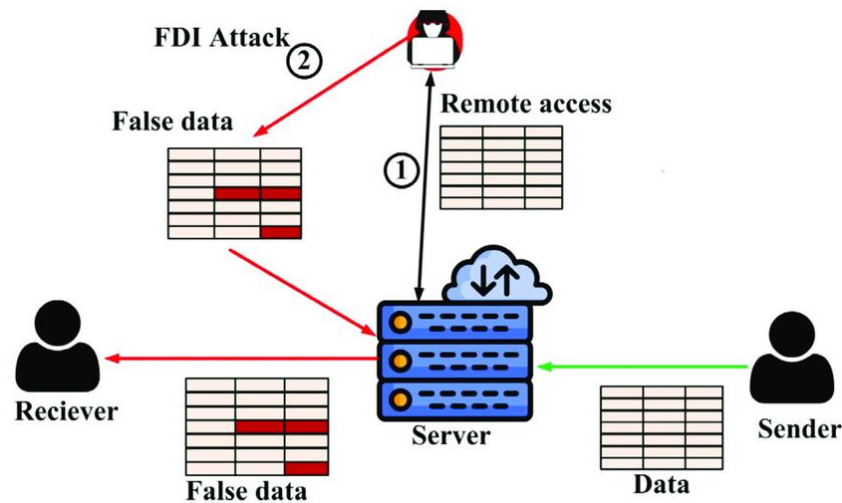


Figure 2.2: Representation of a false data injection attack.

Source: www.researchgate.net¹⁹

Research into detecting such attacks is becoming increasingly relevant, as cybercriminals can use vulnerabilities in communication channels or software to alter consumption patterns.

The main benefits of smart grids include better electricity metering, improved peak load management, reduced losses and integration of renewable energy sources. The key element of smart grids is smart meters, which we will talk about later, but they are often the target of attacks. In general, smart grids are an integrated cyber-physical system whose security, resiliency and reliability depend directly on the correctness and integrity of the collected data. Smart meters play an important role in ensuring energy-efficient and profitable operation of smart grids. However, their use raises some concerns about real-time data collection. For example, the statistics of hourly electricity consumption may include redundant information and the data collection process itself sometimes violates users' privacy. To protect end-user privacy, it is important to introduce more artificial intelligence algorithms and data entanglement techniques that will make smart grids safer. (stellarix.com²⁰)

¹⁹<https://www.researchgate.net/publication/371442754/figure/fig5/AS:11431281166692060@1686330777326/Representation-of-a-false-data-injection-attack.png>

²⁰ <https://stellarix.com/insights/articles/smart-grid-architecture-developments-and-use-cases/>

2.3 Smart meters: architecture, data exchange protocols, vulnerabilities.

The term «smart meter» is most often associated with electric meters, but can also be used for devices measuring gas, water or heat consumption. In general, it is an electronic device that records data on the consumption of resources such as energy, voltage, current and power ratio. Smart accountants provide information to consumers to understand their resource usage behavior, and to providers to monitor the system and bill them. They work almost in real time, transmitting data regularly throughout the day and providing a two-way link between the counter and the central system. (en.wikipedia.org²¹) Smart meters are a key element of Smart Grids, providing automated and detailed collection of data on electricity consumption. As described such devices can collect hourly or even more frequent measurements of consumption, transmit them to the electricity supplier and receive dynamic pricing signals. Smart meter architecture involves the use of built-in microcontrollers, communication modules (such as PLC²², RF²³ or GPRS²⁴) and cryptographic data protection protocols. (cyberleninka.ru²⁵)

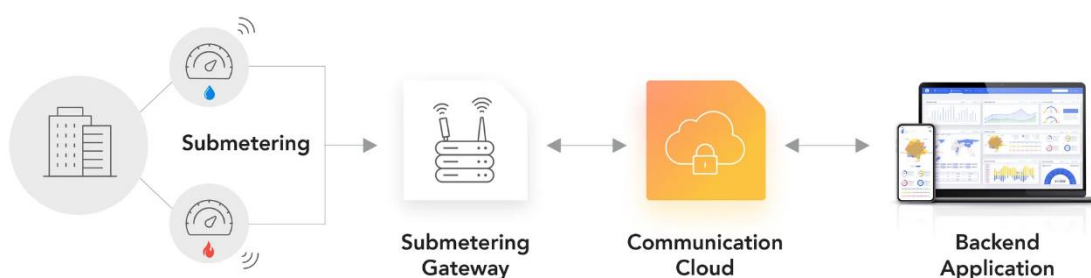


Figure 2.3: How smart meters communicate. Source: www.emnify.com²⁶

Like any IoT (Internet of Things) device, smart meters require a network connection to work. However, there are some common misconceptions about how this connection is made and what technologies are best suited for data transfer to and from such devices. For example, as shown in Figure 2.3 data from smart meter is not always directly transferred to the cloud. Usually they are first sent to a local gateway, which collects

²¹ https://en.wikipedia.org/wiki/Smart_meter

²² Power Line Communication is a time-tested technology that is used for data transmission via electrical wires.

²³ Radio frequency is the frequency of oscillations of an electric or magnetic field, current or voltage in a range from about 20 kHz to 300 GHz.

²⁴ General Packet Radio Service is a wireless technology that provides data transmission through mobile networks.

²⁵ <https://cyberleninka.ru/article/n/obnaruzhenie-anomaliy-setevogo-trafika-metodom-glubokogo-obucheniya>

²⁶ <https://www.emnify.com/hs-fs/hubfs/smart-meter-network-architecture.png?width=2048&name=smart-meter-network-architecture.png>

information from all counters in a certain area. This data is then sent to the cloud where it becomes available to suppliers and customers through dedicated platforms. (David Garcia, 2024)

Modern smart meter systems use Advanced Measurement Infrastructure (AMI) which differs from automatic reading (AMR²⁷) technologies. AMI provides a wider range of functionality, including two-way communication between utilities and consumers, and real-time control allowing more efficient management of power consumption and information exchange. Both AMI and AMR eliminate the need for manual meter reading. However, AMR only provides one-way communication, which limits its functionality. Designing safe and steady smart metering systems involves overcoming logistical and technical difficulties. The main tasks are to define hardware requirements, implement reliable security protocols, maintain user trust and comply with regulatory requirements. These aspects are particularly important in the context of an evolving intelligent infrastructure and are key challenges in deploying such systems. (onomondo.com²⁸)

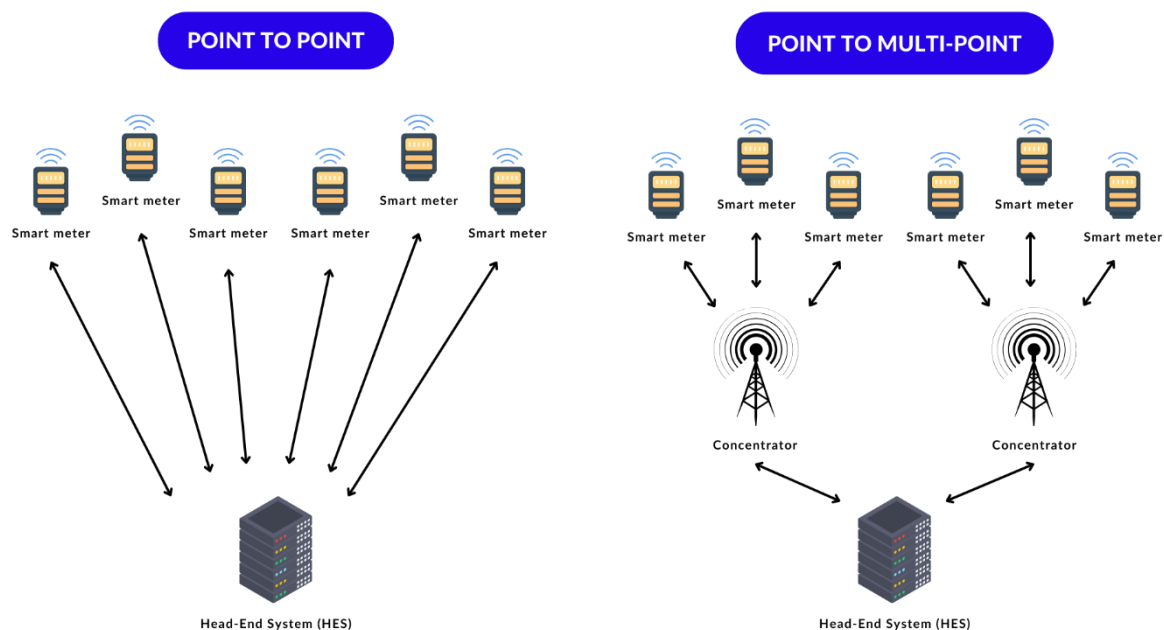


Figure 2.4: Communication technologies. Source: admin.terranovasoftware.eu²⁹

As shown in Figure 2.4 depending on the type of communication used between the meters and the Head-End System(HES), the meters are divided into two major families:

²⁷ Automatic Meter Reading is a technology designed for automatic collection of data on consumption, diagnosis and status of meters of gas, electricity or water.

²⁸ <https://onomondo.com/blog/build-iot-based-smart-metering-systems/>

²⁹ <https://admin.terranovasoftware.eu/img/886f91e9-2276-457f-aef8-249fd6d12207/schema-comunicazione-smart-meter-1.png>

- The first family includes meters with «point-to-point» technology, which establish a direct connection to the Head-End System(HES). In this case, the data exchange takes place within the same session, which makes the synchronous communication.
- Meters with the technology «point-to-multipoint», in contrast, exchange data with the Head-End System(HES) through intermediate devices such as concentrators. The concentrators act as intermediaries, collecting data from several meters and transferring it to HES. This exchange involves requests and responses that are made within multiple communication sessions, making the asynchronous communication.

Another common classification of meter is based on the medium that is used to transmit data packets between the meter and the Head-End System (HES). We have already mentioned them before, and now we will go into more detail. There are four main types of such environments:

- **Power Line Communication (PLC)** - technology that is used exclusively in power smart metering systems. It uses existing electrical cables for data transmission, which makes it convenient to integrate into the power grid.
- **Telephone Line Communication** is a technology that uses traditional telephone lines to transfer data between the meters and the Head-End System.
- **Fiber Optic Communication** is a data technology that uses fibre-optic cables. It provides high speed and reliability of connection.
- **Wireless Communication** is a technology that includes the most popular smart metering solutions such as WiFi, ZigBee and cellular networks. It provides flexibility and ease of installation, making it one of the most common in this field.

In the electric power industry, smart meters have their own unique history. One of the most common communication technologies is Power Line Communication (PLC). This technology is used exclusively in smart electricity metering systems and allows measurements and other data to be transmitted via power lines.

The data transmitted is received at a collection point, which is usually located on the distribution substation supplying the meter and where the concentrator is installed. In some cases, the pick-up point may be located next to the distribution transformer. From these points, data is then sent to the central control systems.

The basic data exchange protocols in smart meters are oriented on reliability and noise resistance in communication networks. However, research shows that there are a number of vulnerabilities. (cyberleninka.ru³⁰)

These include the following:

- Insufficient authentication and encryption of data when exchanging information between the counter and the processing center.
- Possibility of introducing false data (False Data Injection), in which attackers can distort the readings of the instruments.
- Vulnerability of communication protocols that allow remote modification or substitution of data.
- Lack of real-time anomaly detection.

Vulnerabilities of smart meters can be exploited by attackers through attacks targeting the device itself or its interfaces. This may occur in the following ways:

- manipulation of hardware
- changes to firmware or
- use of flaws in system design and implementation.

As for the manipulation of hardware, modern smart meters are designed to continue working correctly even if you lose communication with the central unit or concentrator. This is important for ensuring uninterrupted operation during temporary communication failures. To prevent remote access to the meter, the first step is to protect a wireless module antenna (such as WiFi 802.11 or ZigBee 802.15.4) or use electric filters to suppress high-frequency signals from the power line.

If the attacker has physical access to the counter body, there may be opportunities for accessing firmware, for example through an unprotected ISP port or bypassing security mechanisms. (Carpenter M., 2008)

³⁰ <https://cyberleninka.ru/article/n/obnaruzhenie-anomaliy-v-setevom-trafike-s-ispolzovaniem-metodov-mashinnogo-obucheniya>

Also replacing the meters or their modules with cloned devices, as well as their movement between different places can cause errors in accounting for consumption and payment.

If we talk about Firmware manipulation, then such attacks are aimed at changing the planned execution of the program of the operating system smart-meter. This can be achieved, for example, by interrupting the internal or external power supply or using a local service port. Smart meter manufacturers are actively working on preventing such attacks by implementing security measures, for example, checking the sequence of meters' readings or sending a signal «heartbeats» at certain intervals. However, even such measures do not exclude the possibility of reprogramming the current firmware in memory of smart-meter if the attacker has sufficient insider knowledge and experience. Regarding the Exploiting limitations of design and implementation, even with reliable security concepts and quality system design at the appropriate level, some aspects may remain underdeveloped. This often leads to implementation errors. For example, one of these vulnerabilities is the transmission of encryption keys over unprotected channels, which makes the system vulnerable to attack. (Wright J. Killerbee, 2009)

As a result, smart meters, being the central link of the Smart Grid, need a comprehensive protection system that would include not only reliable cryptographic methods but also algorithms for detecting anomalies, which will be discussed below.

2.4 Methods of anomaly detection: statistical approaches, machine learning, deep learning, hybrid systems.

Anomaly detection is a widely used method to detect unusual patterns in data that are significantly different from normal behavior and may be suspicious. This approach is important because it allows for rapid response to anomalous events, providing information for decision-making and prevention of possible problems. Different methods have been developed and applied in various fields of research. (V. Chandola, A. Banerjee, V. Kumar, 2009)

There is a wide range of methods to detect anomalies in power consumption data. Statistical approaches have traditionally been applied: definition of thresholds, analysis of seasonality, testing of hypotheses on the distribution of data. However, statistical methods may be not flexible enough in complex patterns or when there are several types of anomalies.

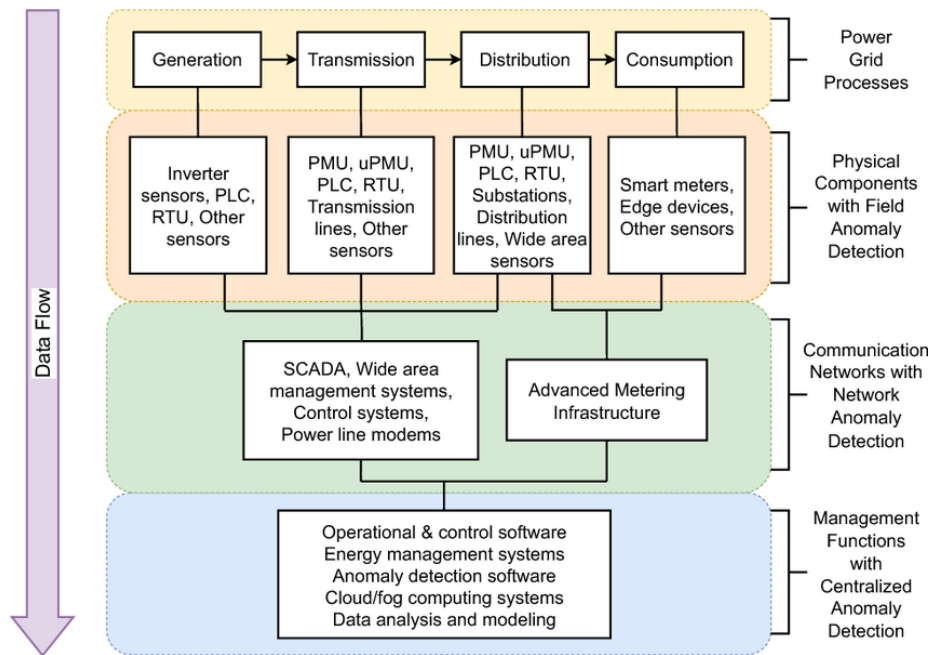


Fig 2.5: Architecture of a power grid with possible applications of anomaly detection at different levels of data flow. Source: www.researchgate.net³¹

Figure 2.5 shows the different types of detection systems that are used depending on the area of monitoring in the power grid. This model is not tied to a specific physical location of anomaly detection, but rather demonstrates different approaches to data analysis. This separation helps to better understand the different methods that can be used by network operators to identify anomalous situations in power systems and combine them effectively into a single monitoring strategy.

Anomalies were first discovered in the industry of telecommunications and industrial systems in the middle of the 20th century by operators who were interested in detecting faults in electrical networks and telecommunication equipment. The main objectives were to minimize system downtime and prevent accidents. The first approaches to detecting anomalies were based on statistical analysis: averages, deviations and threshold values were calculated, the result of which was interpreted as an anomaly.

The growth of computer technologies and increasing data volumes have led to automation challenges, particularly in areas like:

³¹ <https://www.researchgate.net/profile/Srinidhi-Madabhushi/publication/372219473/figure/fig4/AS:11431281192663713@1695780263039/architecture-of-a-power-grid-with-possible-applications-of-anomaly-detection-at-different.png>

- **Financial sector:** detection of fraudulent transactions.
- **Health:** monitoring the condition of patients.
- **Energy:** detection of anomalies in power consumption.
- **Cybersecurity:** detection of network attacks and suspicious traffic.

The detection of anomalies in power consumption data is one of the key methods to detect unusual events in buildings and infrastructure power supply facilities. It helps to detect theft of energy, measurement errors and technical losses. In addition, this approach plays an important role in detecting cyber-attacks in cyber-physical systems such as capacity overload attacks in intelligent networks or microgrids.

In order to prevent power outages, it is important to detect abnormal consumption at an early stage, before it leads to disruptions. This allows for stable operation of electrical networks and equipment, minimizing the risk of power outages.(A. Korba, N. Tamani, Y. Ghamri-Doudane, 2020)

Today, anomaly detection methods are widely used in various industries:

- **Industrial IoT systems:** monitoring equipment to prevent accidents.
- **Financial analysis:** prevention of credit card fraud and risk analysis.
- **Energy:** monitoring of smart grids to detect false data and unauthorized consumption.
- **Medical diagnosis:** detection of anomalies in medical images and patient monitoring data.
- **Cybersecurity:** protection against intrusions and analysis of suspicious behavior.

William Edwards Deming, one of the founders of modern quality management theory, noted:

"In God we trust. All others must bring data." (www.ibm.com³²)

This quote reflects a key principle of working with power grid anomalies: any conclusions should be based on objective data analysis. In modern smart grid systems, the detection of anomalies requires the use of machine learning and statistical analysis techniques to identify hidden patterns and mismatches in power consumption. Without a systematic approach to data collection, processing and interpretation, it is not possible to effectively identify and prevent potential threats such as tampering, hardware failures or fraud.

³² <https://www.ibm.com/think/insights/in-god-we-trust-all-others-must-bring-data>

Thus, a data-driven approach becomes an integral part of ensuring the reliability and security of smart grids.

2.4.1 Statistical approaches to anomaly detection

Methods:

Statistical methods for detecting anomalies are based on analysis of data distribution and detection of deviations from expected patterns. Main approaches include:

- **Mean and standard deviation** - anomalies are determined based on statistical thresholds calculated with respect to the mean and dispersion of data.
- **Time series analysis** - takes into account the seasonal and trend components of the data, allowing significant deviations from typical consumption to be identified. (Box, G. E., Jenkins, G. M., & Reinsel, G. C., 2008)
Time series data is a sequence of measurements performed at equal intervals. This format allows to analyze the dynamics of changes in indicators over time, to identify patterns and also to predict future values. (Lye, J. Hirschberg, 2020)
- **Hypothesis tests** - used to check the compliance of observed values with expected regularities, which helps to identify statistically significant anomalies.
- **Clustering** - method of grouping data by similar features, in which points not belonging to any cluster are considered as potential anomalies.

Benefits:

- Simplicity of implementation and interpretation.
- Efficiency when analyzing small amounts of data.
- Work well in conditions where typical patterns of consumption are known.

Restrictions:

- Low efficiency when working with large and complex multidimensional data.
- Sensitivity to anomalies with complex structure.
- Dependence on assumptions about the distribution of data, which may limit the applicability in real conditions.

Areas of application:

Statistical methods are most effective in monitoring systems where the nature of the anomaly is known in advance and the data volume is relatively small. They can be used, for example, in small businesses or accounting systems where it is necessary to detect sudden variations in power consumption.

2.4.2. Machine-learning techniques of anomaly detection

Machine learning (ML) techniques allow to detect anomalous events in data by analyzing hidden patterns and relationships between features. Unlike statistical methods that are based on predetermined thresholds, machine learning algorithms can adapt to complex multidimensional data and identify complex types of anomalies. They are applied both in teacher-oriented (supervised learning) tasks where there is marked out data about anomalies, and in unsupervised learning, when algorithms detect anomalous patterns themselves.

Methods:

2.4.2.1 Decision Trees & Random Forests:

This method is a hierarchical structure of rules that consistently divides data into classes (normal and anomaly).

- **Decision Trees** - construct binary breakdowns of data based on the most informative features.
- **Random Forests** - an ensemble method that combines several decision trees to increase the accuracy of classification. (Jalilvand, A., Akbari, B., Zare-Mirakabad, F., 2018)

Benefits:

- Easy interpretation of results.
- Ability to work with heterogeneous and multidimensional data.
- High accuracy with marked data.

Restrictions:

- Retrained if the trees are too deep.
- May result in erroneous results if data is not available.

Areas of application:

- Detection of false data in power systems.
- Analysis of power consumption for anomalies.

2.4.2.2 Gradient Boosting (XGBoost, LightGBM, CatBoost)

Gradient boosting is a powerful ensemble method that brings together many weak models (usually decision trees) to improve the accuracy of predictions. It gradually corrects the errors of previous models by teaching each next tree to the errors of the previous one.

Benefits:

- High accuracy and ability to work with large amounts of data.
- It is good at detecting complex patterns.
- Can work with missing values and heterogeneous features.

Restrictions:

- High computational complexity - requires considerable resources.
- Long time learning on big data.

Areas of application:

- Detection of anomaly power consumption.
- Detection of attempted fraud and energy theft.

2.4.2.3 Clustering (e.g., k-medians, DBSCAN, Gaussian Mixture Models)

Clustering methods group objects by similarity. Data that does not belong to any one group can be considered as anomalies.

- **The k-mean algorithm** finds k clusters in data, minimizing the distance between objects within each cluster.
- **DBSCAN** - isolates clusters based on data density and detects emissions as individual points.
- **Gaussian Mixture Model (GMM)** - considers the data as a mixture of several normal distributions and allows anomalies to be determined based on probabilistic estimation.

Benefits:

- Suitable for working with unmarked data.
- Reveals complex patterns in the data.
- Works well in real time.

Restrictions:

- Requires a preliminary determination of the number of clusters (k).
- Sensitive to the scale of features.

Areas of application:

- Classification of users by type of power consumption.
- Identification of atypical consumers and possible manipulations with the meters.

2.4.2.4 Regression models (linear and polynomial regression, neural network approaches)

Regression techniques allow to predict the normal behavior of the system and detect significant deviations that may indicate the existence of an anomaly.

- **Linear regression** - builds a relationship between the input characteristics and the target variable.
- **Polynomial regression** - takes into account non-linear dependencies between variables.
- **Neural networks** are more complex models capable of taking into account non-linear relationships in big data.

Benefits:

- Well predict expected values.
- Work with time series and seasonal fluctuations.

Restrictions:

- Requires proper configuration of parameters.
- Can't always explain why a certain value is considered an anomaly.

Areas of application:

- Forecasting of electricity consumption.
- Detection of sudden changes and deviations from the norms.

Machine learning methods provide powerful tools for detecting anomalies in power systems. Unlike traditional statistical approaches, they allow to work with multidimensional and complex data by identifying patterns that are difficult to detect manually. However, their effective application requires sufficient data volume, quality pre-processing and computational resources. Their integration into smart grids helps to improve system reliability and timely detection of threats such as cyber attacks and fraud.

2.4.3 Deep learning methods for anomaly detection

Deep learning offers powerful methods for detecting complex anomalies that are difficult to detect with traditional statistical and classical machine learning algorithms. Unlike conventional methods, neural networks can automatically find hidden patterns in data, analyze temporal and spatial dependencies, and adapt to changing conditions.

Deep learning techniques are particularly effective when analyzing large amounts of data, such as power consumption time series, network traffic or images. However, their use requires considerable computational resources and careful configuration of the models.

Methods:**2.4.3.1 Recursive neural networks (RNN) and LSTM**

Recursive neural networks (RNN) and their improved version - long short-term memory networks (LSTM) - are designed to work with data sequences and time series. They are able to analyze long-term dependencies, which makes them useful for forecasting and detecting anomalies in power consumption time series.(T. Donkers, B. Loepp, J. Ziegler, 2017)

Benefits:

- Take time dependencies into account in the data.
- Good at analyzing consistent data, such as power consumption or network traffic.

Restrictions:

- Learning difficulties due to the problem of a disappearing gradient (solved with LSTM or GRU).
- High computational complexity.

Areas of application:

- Analysis of time series to detect anomaly power consumption.
- Detection of unusual behavior in power network monitoring systems.
- Detection of malfunctions in industrial systems and prediction of equipment failures.

2.4.3.2 Autoencoders

Autocoders are neural network models that are trained on normal data and then used to detect anomalies. They encode the input data into a compact presentation and then try to recover it. If the difference between input and recovered data is large, the point is considered anomaly. (aiforsocialgood.ca³³)

Benefits:

- Work well without tagged data (analyze normal behavior).
- Can effectively detect rare and complex anomalies.

Restrictions:

- Not always suitable for working with dynamic data.
- Requires careful configuration of the network architecture.

³³ <https://aiforsocialgood.ca/blog/how-artificial-intelligence-is-revolutionizing-various-industries>

Areas of application:

- Detection of a cyber attack in smart grids (FDI-attacks).
- Analysis of power consumption and detection of anomaly jumps or falls.
- Detection of malfunctions in industrial systems.(Hinton, G. E., & Salakhutdinov, R. R., 2006)

2.4.3.3 Transformer-based architecture (e.g., BERT, Time Series Transformer)

Transformer technology, originally developed for natural language processing (NLP), is also used in time series analysis. Unlike RNN, it uses the attention mechanism (Attention), allowing to model complex dependencies in data without losing information about distant events.

Benefits:

- Takes into account the context and complex dependencies in the data.
- Work well with large datasets.

Restrictions:

- Require considerable computing resources.
- Difficult to interpret.

Areas of application:

- Analysis of time series in power systems.
- Identification of long-term dependence on power consumption.
- Detection of complex cyber attacks in smart grids.

2.4.4 Hybrid systems for anomaly detection

Hybrid systems combine the advantages of different approaches, combining statistical methods, machine learning and deep learning to achieve high accuracy anomaly detection. This approach allows to consider both simple statistical patterns and complex dependencies in multidimensional data.

Hybrid models are particularly effective in high data variability, where individual methods can give conflicting results. They help to improve noise resistance, adapt to new conditions and minimize the number of false alarms.

2.4.4.1 Combination of statistical methods and machine learning

This approach involves the use of statistical methods for preliminary analysis of data (e.g., elimination of emissions or normalization) and then applying machine learning models to classify anomalies or combine several ML models to improve accuracy, noise resistance and adaptability to changing operating conditions of the power system.

Benefits:

- Simplicity of interpretation in the first stage.
- Efficiency when limited data is available.

Restrictions:

- May not account for complex relationships in data.
- Limited to statistical model assumptions.

Areas of application:

- Power consumption analysis to detect abnormal jumps.
- Pre-filtering of data before transmission to neural network models.

2.4.4.2 Use of deep learning models to extract features and machine learning for classification

Deep neural networks can automatically extract complex patterns from data and then pass them on to machine learning algorithms such as random forests or gradient boosting for final classification.

Benefits:

- Allows you to analyze complex data structures.
- Improves accuracy by combining powerful methods.

Restrictions:

- Requires a lot of computing resources.
- Complexity of the configuration and interpretation of results.

Areas of application:

- Analysis of time series using LSTM to extract traits and random forests to classify anomalies.
- Detection of cyber attacks in smart grids.

2.4.4.3 Combining several models (ensemble methods)

This method involves using different models to detect anomalies and combining their predictions in order to improve overall accuracy.

Benefits:

- Increased resistance to noise and data variability.
- Reduction of false alarms.

Restrictions:

- The complexity of integrating several models into a single system.
- High demands on computing resources

Areas of application:

- Detection of technical faults in industrial systems.
- Analysis of financial transactions to detect fraud.

CHAPTER 3. METHODOLOGY OF ANOMALY DETECTION

This chapter describes the methodology for anomaly detection used in the thesis. Section 3.1 brief description of methodological approach. Section 3.2 identifies the data sources, detailing their origins, structure, and measurement intervals. Section 3.3 elaborates on the format and characteristics of these datasets, including the volume, feature count, and time series nature of the data. Section 3.4 explains the process of artificially injecting false data (anomalies), detailing various anomaly types . Section 3.5 justifies the selection of specific machine learning models—Decision Trees, Random Forests, Gradient Boosting, and LSTM—based on their applicability to anomaly detection. Section 3.6 describes data preprocessing steps, including data integration, cleaning, transformation, reduction, discretization, sampling and feature engineering. Section 3.7 highlights the metrics chosen for evaluation, such as Precision, Recall, F1-score, and Accuracy. Section 3.8 describes what questions and expectations we have about the result of the thesis and finally, Section 3.9 lists the Python tools and libraries utilized for analysis.

3.1 Methodological approach

The aim of this work is to develop and test a methodology for detecting anomalies in power consumption data using machine learning algorithms. To achieve this goal, the following tasks had to be accomplished:

- Review existing methods for identifying anomalies in power grids and identify their strengths and weaknesses;
- To form a set of data with artificially inserted anomalies, simulating various types of failures and attacks (noise, zeroes, sudden jumps/ drops in consumption, etc.);
- Implement and compare different models of machine learning (decision tree, random forest, gradient boosting, LSTM, etc.);
- Evaluate accuracy, recall, F1-score and other quality indicators for each algorithm;
- Investigate the feature importance and make a visualization of the anomalies found.

The special feature of the approach is the emphasis on the practical applicability of the models, as well as the possibility of their integration into real power network monitoring systems. Aims of this work is bridging the existing gap between theoretical research and application of data analysis technologies to improve the security and resilience of smart meters.

Thus, this methodology allows a system approach to the task of detecting anomalies in the absence of labeled data and source heterogeneity. Each of the steps of the methodology is described in detail in the following sections of the chapter.(Ismeil, M., & Güler, E., 2003)

3.2 Source data sets: where the data, data structure, and timescale (hourly, weekly) came from.

To realize the task of detecting anomalies in power consumption, different datasets were selected, covering a wide range of scenarios: from individual households to industrial enterprises and distribution networks. The choice of different data sources is driven by the need to ensure the universality and scalability of the proposed approach, as well as to test its applicability in different environments.

The main selection criteria for datasets were:

- Temporal nature of records (time series);
- Sufficient detail of measurements (from minute to hourly intervals);
- Availability and openness of sources;
- Representativeness of scenarios of real power consumption;
- The possibility of integrating additional factors (for example, weather conditions, sub-counters, etc.).

The selected datasets cover different geographical regions (North America, Europe, Asia, North Africa), allowing to take into account consumption characteristics depending on infrastructure, climate and type of consumers. Both open-source and industrial data are presented.

Special attention was given to the availability of detailed time-stamping and the possibility of synchronization with other sources (e.g., weather data). This is critical for the correct injection of false data and subsequent model learning.

The table 3.1 below provides a summary of each data set, including source, region, period, measurement interval and brief description.

Nº	Dataset	Source	Region	Interval	Period	Size	Description
1	Network Attacks	IEEE [26]	North America	Hourly	Few months	~100K	Power usage + simulated FDI attacks
2	US Power Data	Kaggle [27]	USA	Hourly	Several years	Millions	Historic regional consumption
3	Tetouan City	UCI [28]	Morocco	Hourly	two months	~7K	Residential + weather info
4	AEP Consumption	Kaggle [29]	USA	Hourly	2004–2018	~120K	Hourly data from AEP
5	Steel Plant	Kaggle [30]	India	Minutes	four months	~50K	Industrial data with sensors
6	Household Power	UCI [31]	France	Minutes	2006–2010	~2M	Household usage, 3 zones

Table 3.1

3.3 Format and characteristics of the source data: volume, number of features, time series.

The raw data used to analyze power consumption is usually presented in CSV (Comma-Separated Values) format, where each row corresponds to a certain time interval. Table data is the most convenient and intuitively understandable format for entering into machine learning models. In such a structure, each string represents a separate object or observation, and each column is a feature (thread) or target variable. However, data for analysis may also come from other sources such as log files, protocol buffers or event streams. These formats may require additional pre-processing and conversion to a tabular view before use in machine learning models.

Features in machine learning are data that is used as input values for the model. They are necessary both in the training stage of the model and during predictions (inferences), because it is the signs that transmit information to the algorithm from which it draws conclusions. Features can be very different:

- Numeric (e.g., age, price, temperature).
- Categorical (e.g., hair color, preferred film genre).

Features are the basis for the work of machine learning algorithms, because they carry the key information needed to build quality models. Their correct selection and processing is crucial to improve the accuracy of predictions and model performance in various fields. (www.hopsworks.ai³⁴)

The main feature is the amount of power consumed (for example, in kilowatt-hours) during a given period (hour, day and so on. d.). Additional fields may include:

- **Time stamps** (date, hour, minute) that allow you to link the record to a specific moment.
- **The user or measuring device ID**, which allows to track data on specific users or network nodes.
- **Meteorological parameters** (temperature, humidity, solar radiation) that can have a significant impact on the level of power consumption.
- Contextual features such as:
 - **Type of day** (working/weekend), because the electricity consumption on weekends can be significantly different from the daily ones.
 - **Seasonal factor** (winter, summer) to account for changes in load in different periods of the year.
 - **Tariff periods**, as the price of electricity may depend on the time of day (for example, daytime and night tariffs).

The number of features can range from a few (in simple data sets) to dozens (in more detailed samples with multiple contextual characteristics).

The sample size can vary significantly depending on:

- **Measurement frequencies** (e.g., hourly or daily data).
- **Duration of observations** (several months, years).
- **Number of objects monitored** (households, enterprises, urban areas).

Data volume is the total amount of information that is stored, processed and transmitted in the system. It can be measured in bytes (kilobytes, megabytes, gigabytes, terabytes and more) or in the number of records (e.g., database lines or files).

Data volume has a significant influence on data management, defining storage requirements, computing resources and processing complexity. The more data, the higher the storage requirements, processing speed and efficiency of analysis algorithms.

³⁴ <https://www.hopsworks.ai/dictionary/feature-data>

Data monitoring platforms allow to detect unusual patterns or significant deviations in the amount of data. This helps to detect potential data quality issues that may occur for various reasons, such as technical failures, integration errors or business process changes. For example, a sudden increase in the volume of a log file may indicate a system error, and an unexpected surge of user activity may indicate a possible anomaly behavior or system crash. The detection of such deviations allows to react quickly to problems and ensure data reliability. (dqops.com³⁵)

Data volumes can range from tens of thousands to millions of records, depending on the duration of observations and frequency of measurements. The availability of large data with a wide range of characteristics allows for more reliable testing of anomaly detection methods and increases the accuracy of detecting power consumption deviations. (Appel, K. W., Gilliam, R. C., Davis, N., Zubrow, A., Howard, S. C., 2011)

Time series are a sequence of measurements or observations recorded at certain points in time. In its simplest form, such data consists of the value of the indicator and the time when it was measured. Despite the simplicity of the concept, time series play an important role in data analysis and are widely used in various industries including energy, finance and industry.

Time series data are valuable for analysing trends, identifying patterns and determining relationships between different indicators over time. The ability to monitor how a parameter changes over a certain period of time gives a deep understanding of processes in different areas - from business analytics to monitoring the status of industrial equipment.

In addition, with the development of technology, the reduction of communication costs and the proliferation of smart devices and sensors, the volume of time series data continues to grow rapidly, making them increasingly important for analysis and decision-making.

³⁵ <https://dqops.com/what-is-data-volume/>

- **Timestamp:** The generation of time series data can occur on a predetermined timer or in response to a certain event. Each record of such data is always accompanied by a time stamp, which allows to organize and index the data by time. It is this time stamp that serves as the key parameter for performing calculations and analysis, ensuring accurate tracking of changes in indicators over time. (Jeff Tao, 2023)

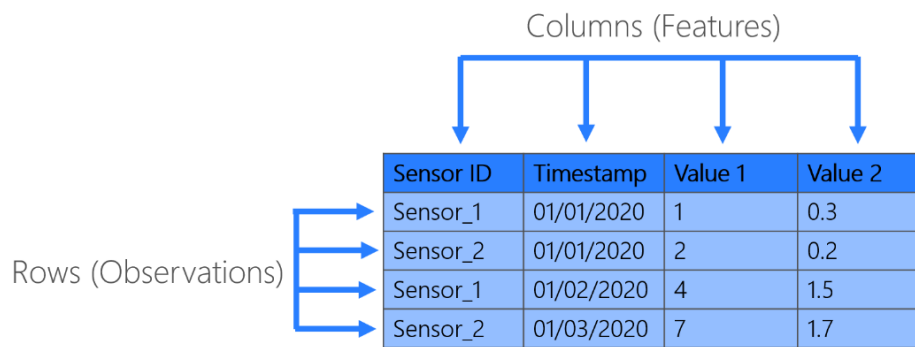


Figure 3.1: Observations versus features in a time series dataframe.

Source: miro.medium.com³⁶

In Figure 3.1 data structure is typical for time series, where each record contains indicators linked to a time point. Data in this format can be used for analysis, prediction and anomaly detection in monitoring systems, intelligent power grids and other areas related to sensor measurements.

As a rough rule of thumb, the machine learning model should be taught at least on order (or better by two) more examples than the number of its learning parameters. However, in practice, quality models usually require much more data. Models trained on large data sets with few features tend to show better results than models using small samples with many features. Historically, Google and other companies have achieved great results by teaching relatively simple models on large-scale datasets. The amount of data needed to build an effective model can vary greatly depending on the particular task. In some cases, a few dozen examples are sufficient to produce satisfactory results, whereas for more complex tasks even trillions of examples may be insufficient. If the data is not sufficient, a good alternative is to use transfer learning, in which the already trained big data model adapts to the new task, allowing high accuracy even with a small set of examples. (developers.google.com³⁷)

3.4 False data injection process and types of anomaly

In order to model different scenarios of power consumption disturbances, the artificial injection of anomalies known as False Data Injection into the original data was implemented as part of this study. One of the key steps in building a system to detect anomalies in power consumption is to model attacks or malfunctions by injecting false data injection. This approach allows to create synthetic scenarios in which the behavior of the system deviates from the norm, and thus test the stability and sensitivity of the algorithms of machine learning to different types of disturbances. The goal of this step is to simulate potential attacks or technical failures that can affect the smart meter readings and the behavior of the system as a whole. As real anomalies are rare and irregular, the technique of False Data Injection is used, in previous chapter was discussed too.

The injection method is aimed at simulating the actions of the attacker or technical failure, which can affect the readings of smart meters. This is especially important for critical infrastructure such as power grids, where incorrect data can lead to significant financial and operational consequences. The use of artificially created anomalies allows:

- Test the model's ability to distinguish between normal and anomaly patterns;
- Evaluate the resistance of algorithms to different types of interventions;
- Train the model in a balanced set of classes (norm and anomaly).

The process of artificially adding anomalies to data involves several steps:

1. *Select the source data set to be used for testing.*
2. *Identify the types of anomalies that need to be entered into the data (for example, consumption spikes or noise distortions).*
3. *Introduction of anomalies in the data taking into account their characteristics (intensity, frequency, duration).*
4. *Evaluation of the impact of anomalies - analyzing how algorithms react to introduced distortions and how accurately they recognize them.*

In this work, several types of anomalies corresponding to a previously defined classification are introduced into the source data:

- **A1: thr (threshold)** - The substitution of values that go beyond a reasonable consumption, for example a sharp increase in consumption.
- **A2: noise (noise)** - adding random deviations from the real values that mimics sensor interference or error.

- **A3: zero (zeroes)** - replacement of normal values with zero, where logically the system cannot have zeroes, which can mean a system failure or equipment shutdown.
- **A4: load-decrease (reduction of load)** - artificial reduction of the consumption level for a certain interval, such as in case of device malfunction or manipulation of data.
- **A5: load-increase (increase in load)** - artificial increase of the consumption level, which can signal an overload of the system or external interference.
- **A6: repeat (repeating patterns)** - copying a certain time interval and repeating it to mask real changes, which may be a sign of data collection error or information substitution.

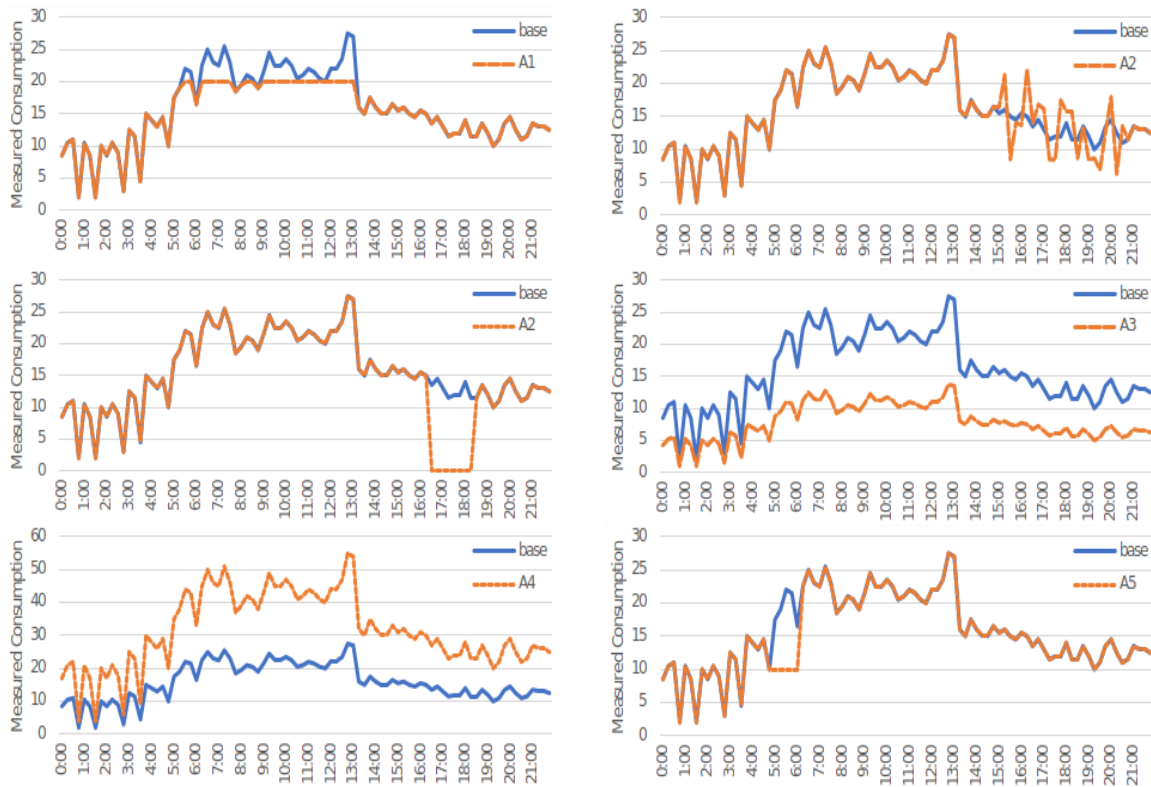


Figure 3.2: Effects of anomalies A1 – A6 to consumption detectors (orange dashed) with respect to normal data (blue solid line). From the top left to the bottom right: A1 anomaly, A2 anomaly, A3 anomaly, A4 anomaly, A5 anomaly, A6 anomaly.

Source: T. Zoppi, I. Bicchierai, F. Brancati, A. Bondavalli and H. -P. Schwefel, 2024

Figure 3.2 shows the behaviour of each of the six anomalies compared to normal consumption. The blue line is normal, the orange dot is distorted. This graphical representation helps to visually assess how strong the anomalies are and how far they can be detected by the model.

Each of these types of anomalies can simulate real situations occurring in power systems and be used to test detection algorithms.

To evaluate the performance of false data detection methods, a clear definition of the types of anomalies is required. In research practice and applied tasks related to power consumption, several characteristic types of distortions are distinguished. Energy systems face different types of data distortions, which can occur both naturally and intentionally. These distortions complicate the analysis and can damage the stability of the power systems. Each type of distortion, its origin, impact and possible threats are described in detail below.

1. Threshold anomalies (thr):

Threshold anomalies occur when power consumption values exceed established limits, which are determined by historical data or norms. For example, a sudden increase in electricity consumption at night in a residential building can signal an anomaly.

- **Reasons:**
 - Technical failures such as measurement device malfunctions or sensor overload.
 - Influence of weather conditions such as high heat or cold that cause increased use of heating or air conditioning.
 - Deliberate actions of intruders who introduce false data to manipulate forecasts or prices for electricity.
- **Examples of locations:**
 - Residential buildings where devices may not correctly record consumption peaks.
 - Industrial sites with sudden load changes.
- **Vulnerabilities:**
 - Failure of power consumption forecasting algorithms.
 - Errors in the calculation of the grid balance, which can lead to network overloading or inefficient allocation of resources.

2. Noise anomalies (noise):

Noise anomalies are the addition of random deviations that distort real data. These distortions can mask more serious problems or make analysis difficult.

- **Reasons:**
 - Electromagnetic interference in communication lines.

- Malfunctions in data collection devices, for example due to old or damaged equipment.
- Hacking attacks that create artificial noise to hide false data or carry out complex attacks.
- **Examples of locations:**
 - Data transfer systems between smart meters and central servers.
 - Central databases where the data is aggregated.
- **Vulnerabilities:**
 - Decrease in forecast accuracy.
 - Deterioration of monitoring systems due to difficulties in filtering and analysing data.

3. Zero (zero):

This type of distortion occurs when zeros are recorded instead of actual data, which is not possible in situations where power consumption should be nonzero.

- **Reasons:**
 - Errors in connection between the meter and the data processing server.
 - Deliberate interference to disguise power consumption, for example, to avoid payment.
 - System failures or software errors.
- **Examples of locations:**
 - Smart meters installed in homes and businesses.
 - Central data stores where data can be corrupted.
- **Vulnerabilities:**
 - Loss of reliable data.
 - Failure of analytical systems, which leads to errors in calculation of tariffs or energy balance.

4. Load-increase/load-decrease (LGA):

Unrealistic changes in load (sharp increases or decreases) often have no apparent cause and may be the result of intervention.

- **Reasons:**
 - Temporary weather changes that increase or decrease consumption, such as seasonal spikes in air conditioning use.
 - Malicious intervention to reduce electricity costs or impact on the market.

- **Examples of locations:**
 - Industrial enterprises where the reduction of load may be associated with an attempt to hide the actual use of electricity.
 - Urban distribution networks exposed to cyber attacks.
- **Vulnerabilities:**
 - Distortion of market data leading to incorrect tariffs.
 - Overloading or unsustainable use of power system resources.

5. Repeating patterns (repeat):

Distortions of this type manifest themselves in the form of artificially created repetitive data that mask real changes.

- **Reasons:**
 - Software errors that cause data to loop.
 - The deliberate introduction of repetitive data to hide changes, for example in the case of electricity theft.
- **Examples of locations:**
 - Power grid database, where consumption time series are stored.
 - Smart meter systems, where data can be distorted during the collection stage.
- **Vulnerabilities:**
 - Masking real data, which makes it difficult to identify violations.
 - Deterioration in the effectiveness of analytical systems working with time series.

Such anomalies can be the result of technical failures or deliberate cyberattacks. Their correct identification is a key task to ensure data integrity and efficient operation of analytical systems.

This process allows to test and configure the anomaly detection algorithms in conditions that are as close as possible to the real, which makes them more reliable and accurate. This process is done by modifying the source CSV file. Some data remains «clean» and some is distorted according to certain scenarios of anomalies. Thus, we know in advance where and what type of anomaly is inserted, which allows to objectively evaluate the work of detection algorithms.

In real data, anomalies are rarely found, which makes it difficult to learn and test the algorithms for their detection. Artificial injection of anomalies allows to create controlled scenarios and test how effectively models can detect deviations. This approach is important for calibrating detection methods and adapting algorithms to different situations that may arise in power systems.

Main purposes of the anomaly injection:

- Evaluation of the efficiency of anomaly detection algorithms.
- Improved adaptation of models to different types of anomalous behavior.
- Simulation of real threats and testing the system for failure resistance.

When artificially injecting anomalies it is important to consider several factors:

- **Reliability** - artificially created anomalies must conform to real scenarios, otherwise algorithms can adapt to unplausible deviations and lose effectiveness on real data.
- **Risk of re-training** - if the model is trained only on artificial anomalies, its effectiveness in working with real deviations may be reduced.
- **Diversity of data** - algorithms should be tested on different types of anomalies, in order not to be limited to the detection of only one type of abnormality.

The method of artificial injection of anomalies is used not only in the field of energy consumption analysis, but also in other areas:

- **Cybersecurity** - simulation of attacks with the introduction of false data into smart grids.
- **Financial sector** - creating artificial fraudulent transactions to train detection systems.
- **Industry** - Equipment failure modelling for fracture prediction.

The application of this method allows to create more reliable anomaly detection algorithms that are able to work effectively in real conditions, reducing the risks of failures and losses.

3.5 Model and Algorithm Selection: Decision Tree, Random Forest, Gradient Boosting, and Neural Networks.

In this work, several machine learning algorithms were selected, which showed high efficiency in the task of detecting anomalies in time series data. The main selection criteria were the algorithms' ability to work with unlabeled or partially labeled data, noise resistance, scalability, and successful application of these methods in related studies, which was discussed in section 1.5.

The following algorithms were used in the study:

1. Decision Tree:

Decision trees are one of the basic, but also efficient algorithms for classification and regression. Their advantage is high interpretability, since the model builds a sequence of logical conditions by which decision-making takes place.

Why is it chosen:

- The decision tree was chosen as the basic model because of its simplicity, interpretability and ability to quickly obtain a primary assessment of the data structure. It serves as a starting point for comparison with more sophisticated models.

Advantages(Benefits):

- Simplicity of implementation and understanding;
- Ability to work with numerical and categorical features;
- Does not require data normalization.

2. Random Forest:

The algorithm is based on a set of decision trees trained on different data and features. The final decision is made by voting (in classification) or averaging (in regression).

Why is it chosen:

- The Random Forest model was chosen for its resistance to re-training and high accuracy on tabular data. It is able to detect complex relationships between features and at the same time provides an embedded assessment of the importance of features.(Tuganishuri, J., & Tuganishuri, J., 2023)

Benefits:

- High accuracy;
- Resilience to retraining;
- Ability to identify important signs.

Random Forest is a reliable basic solution in anomaly detection tasks.

3. Gradient Boosting:

An assembly method in which models are sequentially trained on previous errors. Each following model tries to correct the errors of previous ones, improving the final quality.

Why is it chosen:

- Gradient boosting shows excellent results on classification and regression problems in tabular data conditions. It is able to effectively deal with emissions and complex non-linear dependencies between features, which are important for power consumption analysis.

Benefits:

- High accuracy;
- Emission resistance;
- Works well with small data sets.

Gradient Boosting provides high quality on table data with a large number of complex interactions between features.

4. XGBoost (Extreme Gradient Boosting):

XGBoost is a gradient boost implemented with an emphasis on performance and scalability. It is often used in data analysis competitions (such as Kaggle) and shows high results on tabular datasets.(xgboost.readthedocs.io³⁸)

Why is it chosen:

- XGBoost shows outstanding results on tabular data and is considered one of the most powerful boosting algorithms. It has built-in regularization mechanisms and high learning speed.

Benefits:

- Working with missing values;
- Resistance to multicollinearity;
- Flexible system of parameters for model customization.

XGBoost works particularly well in tasks where it is important to capture non-linear dependencies and interactions between features.(Fang, W., Liu, Y., Xu, C., Luo, X., Wang, K., & Wang, K., 2024)

5. LSTM:

³⁸ <https://xgboost.readthedocs.io/en/stable/>

A recursive neural network designed to work with sequential data. LSTM remembers long-term dependencies and can take into account the context of previous states.

Why is it chosen:

- LSTM was chosen as a model capable of taking into account the time dependencies between observations. Unlike previous models, it "remembers" the context and can capture long-term patterns characteristic of energy time series (e.g., daily, weekly, seasonal cycles).

Benefits:

- Ability to model temporal dependencies;
- Resistance to disappearing gradients;
- Suitable for pattern and trend analysis.

LSTM is used in this study as a specialized model for time series, providing a deep context in the analysis of power consumption.

Limitations:

- **Lack of labeled data:** the absence of precise markers of anomalies limits the use of fully controlled learning.
- **Data volume for neural networks:** models such as LSTM require large amounts of data and careful hyperparameter tuning to achieve stable results.
- **Time cost of teaching the ensembles:** Gradient-boosting models can be taught considerably longer, especially when working with large datasets.
- **Generalizability:** some algorithms, especially decision trees, can be poorly adapted to new data types without re-training.
- **Interpretability:** while trees and random forests are easily interpreted, LSTM and boosting require additional methods of explanation (e.g., SHAP).

3.6 Data Pre-processing and Feature Engineering

Pre-processing of data is an important step in the analysis of time series, because the quality of the prepared data depends on the accuracy of the algorithms of machine learning and anomaly detection. Data from different sources may include noise, missed values, emissions and other artifacts that need to be eliminated before analysis. Pre-processing is the process of preparing raw data for further analysis. It involves organizing the data, converting them to a convenient format and eliminating errors. At this stage, the non-deleted or duplicated records are removed, the missing values are filled in and inconsistencies and possible errors are corrected.

The main objective of this process is to make the data accurate, complete and ready for application of analytical methods and machine learning algorithms. Data cleaning and preprocessing usually involve several key steps that help to improve the quality of the data and minimize the impact of noise on the analysis results. Before using the anomaly detection methods, data is subjected to a series of pre-processing steps:

- **Data integration** is the process of combining information from different sources into a single, consistent set of data. This step is necessary to create a coherent and consistent data structure that will allow efficient analysis, identification of patterns and application of machine learning algorithms. Integration can include combining data from various sensors, databases, file stores or cloud services to provide a more complete understanding of the system and its behavior.
- **Data cleaning** is a step where the data is checked for errors, inconsistencies and anomalies and then corrective actions are taken to correct them. This process is necessary to ensure the accuracy and reliability of data before its analysis. Data cleaning plays a key role in producing high quality input material for machine learning and analysis, allowing to minimize the impact of incorrect data on final results. The main tasks of data cleaning include:
 - **Duplicate deletion** - elimination of duplicated records that may distort the results of the analysis.
 - **Missing data processing** - replacing missing values with mean, median or predicted values based on other data attributes.
 - **Correction of formatting errors** - bringing the data to a single format (for example, unification of dates, units of measure, case).
 - **Emission filtering** - the detection and elimination of abnormally high or low values caused by measurement errors or system failures.

- **Data transformation** is a step in the preparation of the data for further analysis, during which they are converted into a user-friendly format. This process is necessary because data from different sources can have different formats, structures and units of measurement, which makes it difficult to analyse and use in machine learning models. Correct data transformation plays an important role in increasing the efficiency of machine learning algorithms, allowing models to better understand patterns in data and make more accurate predictions. The main operations at this stage include:
 - **Data type conversion** - converting data to the correct formats (for example, converting rows to numeric values or dates).
 - **Data scaling** - normalization of numerical features to bring them to a common range of values (for example, Min-Max normalization or standardization).
 - **Encoding of categorical variables** - conversion of categorical data into a numeric format (for example, by One-Hot Encoding or Label Encoding methods).
 - **New feature generation** - creation of additional variables based on existing data (for example, the selection of time characteristics from time tags).
- **Data reduction** is the process of reducing the amount of information in a dataset by selecting the most important features or reducing the dimension. This step is especially important if the source data set contains a large number of features, many of which may be redundant or irrelevant for analysis. The use of data reduction techniques allows to reduce computational complexity of analysis, improve performance of algorithms and increase model interpretability while preserving key patterns in the data. The main methods of data reduction include:
 - **Feature selection** - selection of the most relevant variables for analysis using statistical methods, correlation analysis or machine learning algorithms.
 - **Dimensionality reduction techniques** - such as principal component analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE), which help to reduce the number of features while preserving basic information.

- **Removal of excessive or weak features** - the exclusion of variables that contain many missing values, have low variability or strongly correlate with other features.
- **Data discretization** is the process of converting continuous numerical values into discrete categories or ranges. This approach facilitates data analysis, makes it more interpretable and can improve the performance of machine learning algorithms, especially in classification tasks. Discretization is particularly useful in the analysis of time series, financial data and power consumption where it allows to identify patterns and facilitates interpretation of results. The main methods of data discretization include:
 - **Equal-Width Binning (EWB)** - to break the range of values into spaces of equal width.
 - **Equal-Frequency Binning (EGF)** - divide data into groups so that each category has approximately the same number of values.
 - **Clustering** (e.g., k-middle) - grouping of values based on their similarity.
 - **Methods based on entropy** (for example, Decision Tree Binning) - determination of the boundaries of intervals based on information measure (more often used in classification algorithms).
- **Data sampling** is the process of selecting a subset of data from the total volume, when a full analysis of the entire data set is not possible due to its large size. The main objective of the sample is to reduce the computational costs while preserving the key characteristics of the source data. Data sampling is an important tool in situations where the complete set of data is too large to be processed, but it is necessary to preserve the representativeness of the sample so that the analysis results remain reliable:
 - **Random Sampling** - selection of random records from the dataset, which allows to obtain a representative subset.
 - **Stratified Sampling** - split data into groups (strata) and proportional selection of records from each group to keep the feature distribution.
 - **Systematic Sampling** - is a sample of records at equal intervals (for example, each n-th row in the dataset).
 - **Cluster Sampling** - split data into clusters and random selection of some clusters for analysis.(openstax.org)³⁹

The first step in data processing is **data cleaning**. From a theoretical point of view, several studies focus on the removal of missed values, emissions and incorrect records. Statistical criteria, such as the three-sigma rule or the inter-apartment range (IQR), are used to identify anomalous values that may distort the model. In addition, studies show that the elimination of noise data and duplicates contributes to the improvement of the model's generalizability. In practice, during the data preparation process, columns that do not contain useful information for model training (such as timestamp and seq_name) were removed, which allowed to minimize the impact of unnecessary noise. Missing values and duplicates were also checked, and standard methods were used to remove them when necessary. In particular, all null-values in the data were replaced by 0 using the `fillna()` method⁴⁰, which allowed to avoid potential errors in data processing and model training.

Normalization of data is the next important step in pre-processing. From the theoretical point of view, bringing features to a single scale is critical for the correct operation of many machine learning algorithms. This is especially true for neural networks, k-nearest neighbors (kNN) and support vectors (SVM), where differences in the scales of input traits can lead to their uneven influence on the resulting model. Various methods of normalization are considered in the literature:

- **MinMaxScaler** - converts the values to a specified range (usually [0,1]) while preserving the distribution of data.
- **StandardScaler** - returns the data to an average value of 0 and a standard deviation of 1, which makes them more convenient for algorithms that are sensitive to feature distribution.

The choice of normalization method depends on the nature of the data and the requirements of the model. MinMaxScaler was used in the scope of this implementation, which allowed to bring all input data to a single range and reduce the influence of the features that dominate by scale. This approach helps to improve the consistency of the model and increase the accuracy of predictions. (Mang, L. D., 2024)

⁴⁰ https://www.w3schools.com/python/pandas/ref_df_fillna.asp

Aggregation of data allows to combine information to reduce noise and identify stable patterns in the data. Theoretically, this process is based on the application of statistical measures such as mean, sum, minimum, maximum and standard deviation to grouped data within set time intervals. Sliding window methods are widely used in time series studies, which allow to calculate statistical characteristics for each time period and to identify local patterns. Time-based data grouping (e.g., hours, days, weeks) is also used to analyse seasonal patterns and long-term trends. In practice, the aggregation was carried out at the selection stage. The non-recurring columns were removed, and if there was a temporal component, data could be aggregated to calculate statistical characteristics. This approach has made it possible to highlight important patterns and reduce the impact of random fluctuations in data.

Time-based feature extraction plays a key role in the analysis of time series because it allows models to take into account cyclicity, trends and seasonal fluctuations. Time characteristics help to improve the predictive capabilities of models, especially in the task of forecasting and anomaly detection. From a theoretical point of view, temporal features may include:

- **Categorical variables** - day of the week, month, hour of the day, season.
- **Numeric variables** - number of days from the beginning of the year, ordinal number of the week.
- **Time series decomposition methods** - allow to highlight trend and seasonal components of time data.

In the process under consideration, the timestamp column was removed at the feature selection stage because it did not carry any relevant information in its original form. However, if the time information were useful for analysis, it could be converted into individual features such as day of the week, hour of the day or weekend indicator that would help models better capture seasonal and time dependencies.

The statistical feature extraction aims to obtain key data distribution characteristics that help models better understand variability and patterns in data. This step is particularly important when analyzing time series and detecting anomalies, as statistical indicators allow to detect deviations from normal behavior. In theoretical terms, the main statistical subjects include:

- **Mean** - the total level of the measured value.
- **Median** - a central trend indicator, emission-resistant.

- **Variance and standard deviation** - the degree of variation of data relative to the mean value.
- **Skewness** - the degree of deviation of the distribution from the symmetrical form.
- **Kurtosis** - the characteristic of the distribution form, indicating the presence of sharp or slight peaks.

These statistical characteristics can be computed both across the whole data set and locally using sliding windows, which allows to analyze the dynamics of changes over time. In this implementation the explicit selection of statistical features was not carried out, but this step is often used to enrich the model information space. The addition of such features can improve the quality of prediction and detection of anomalies, especially in time analysis tasks where it is important to consider not only the values of the indicators but also their changes and distribution over time.

Data splitting is one of the key steps in building and evaluating machine learning models. Its main goal is to provide an objective verification of the model's ability to generalize knowledge and correctly work with new, previously unknown data. Without a correct breakdown of the data, there is a risk of relearning the model or, conversely, its insufficient adaptation to real scenarios.

- **Train Set** contains the main data on which the model learns, detecting patterns, connections and dependencies. At this stage, the algorithm determines the internal parameters of the model (e.g., weights in the neural network or coefficients in the linear regression) that are used to form predictions.
- **Validation Set** is used to adjust and optimize the model's hyperparameters. Hyperparameters are parameters that do not change during the learning process of the model, but have a significant effect on its quality (for example, depth of the decision tree, regularization coefficients or number of neurons in hidden layers of the neural network). Validation set allows to track whether the model is starting to retrain, that is to say to remember specific examples instead of identifying general patterns. If you only use a training set when configuring the model, there is a risk of creating an algorithm that shows high results on familiar data but does not work well on new examples. (Suhartono, S., Amalia, F. F., Saputri, P. D., Rahayu, S. P., & Ulama, B. S. S., 2018)

- **Test Set** is intended for final independent model verification. It is not involved in the training and selection of hyperparameters, but is used exclusively at the final stage for objective evaluation of the quality of the model. Checking on train set allows to determine how well the model is able to work with real data. If the accuracy is significantly lower than that of the training and validation sets, this may indicate problems with the generalization capability of the model, making it less suitable for practical use. (Emmert-Streib, F., & Dehmer, M., 2019)

It is important to note that in some cases sets are formed taking into account the time scale. For example, when predicting time series, earlier data is sent to the training and validation sets, and later segments are highlighted in a test set, which simulates the real situation where you need to predict the future based on information from the past. In addition, cross-validation is often used for a more accurate assessment of the quality of the model. In this method, the training set is split into several subsets, and the model is learned multiple times by alternating training and validation sets. The final test set remains unchanged until the model is finally checked. This approach allows to reduce the dependence of the model on a specific breakdown of data and to obtain a more objective evaluation of its quality.

In this work, data were divided into training and test sets at the ratio of 80/20, which allowed to correctly assess the quality of the model. Additionally, a validation set was selected which was used to adjust the hyperparameters and prevent re-training. This approach has made the model more stable and reliable on new data.

Based on the theoretical basis, a final processing of the initial data set was carried out to prepare for the training of the machine learning model. The timestamp and seq_name columns were first removed from the dataset because they did not provide useful information for the model and could cause excessive noise. After this, a clean set of attributes was formed and the label column was left as the target variable.

Since most machine learning algorithms require working with numeric attributes, LabelEncoder was used to convert string labels into a numeric format. This ensured a correct interpretation of the categorical data in the model. To bring all input data to a single scale, MinMaxScaler was applied, which normalized the signs in the range [0, 1]. This avoids the situation where features with different scales have an uneven effect on the model.

Data was split into training, validation and test sets. This step was necessary for:

- Optimization of the model's hyperparameters.
- Prevention of retraining.
- Final independent quality check of the model on new data.

Since the model used a recursive neural network (LSTM), the data was converted to a three-dimensional format (samples, timesteps, features), where the timesteps parameter was set to 1. This is necessary for correct processing of time sequences and taking into account the time dependency in the data.

The pre-processing stages allowed to create a clean and balanced data set, ensuring correct learning of the model and an objective evaluation of its quality.

Feature engineering is an important step in the process of machine learning, which directly affects the quality of classification and the ability of the model to distinguish between normal and abnormal patterns. This section considers both general approaches to feature engineering for time series and specific solutions used in this work.

General approaches to the formation of features in the tasks of analysis of time series:

- **Temporal Features:**
 - Hour of the day, day of the week, day of the month, month - used to account for seasonality and repeated patterns.
 - The features «day off/working day», «holiday» etc. - are important for analysis of human activity affecting power consumption.
- **Lag Features:**
 - The values of features in previous moments (for example, 1, 2, 3 hours ago).
 - Used to model the time dependence in data.
- **Rolling Statistics:**
 - Moving average, median, standard deviation, minimum/maximum for N previous steps.
 - Help identify trends, local spikes and instability.
- **Derivative (Differencing & Rate of Change):**
 - The difference between current and previous values (delta).
 - Acceleration (second derivative), current to previous ratio, etc.
- **Categorical Features (Calendar-based Encodings):**
 - The day of the week as a one-hot-coding or cyclic representation (using sine and cosine).
- **Statistical normalization:**

- Min-Max Scaling, Standard Scaling, Robust Scaling - especially important when working with scale-sensitive models.

Implemented features:

- **Laggy feature (lag = 1):**
 - Feature added, which reflects the value of consumption in the previous time step. It helps models to detect changes in time and take into account recent behavior.
- **Feature delta:**
 - Calculated as the difference between current and previous values of power consumption:

$$\text{delta} = \text{current consumption} - \text{previous consumption}$$

- The feature was used to fix sudden jumps and falls. However, during the experiments it was found that the inclusion of delta does not lead to an improvement in classification quality.
- **Moving averages (Moving Average):**
 - To smooth the time series and identify stable trends, two features have been added:
 - MA(3)** is the average value for the last three time steps;
 - MA(5)** is the average of the last 5 time steps.
 - These features contributed to the reduction of the sensitivity of the model to minor fluctuations and noise in the data.
- **Normalization of data:**
 - All the features were normalized using Min-Max Scaling up to [0, 1] range for correct operation of the models.
- **Data separation:**
 - The data were divided into a training (70%), validation (15%) and test (15%) sample randomly, with proportions between normal and abnormal marks.

This set of features proved to be sufficient for the successful detection of artificially introduced anomalies and corresponded to the proposed experimental objectives.

3.7 Metric selection: Precision, Recall, F1-score, Confusion Matrix, Accuracy evaluation.

Evaluation of the quality of the machine learning model plays an important role in analyzing its effectiveness. For a correct comparison of different approaches to detecting anomalies and correct interpretation of the model's work, different metrics are used, each of which evaluates a certain aspect of its effectiveness.

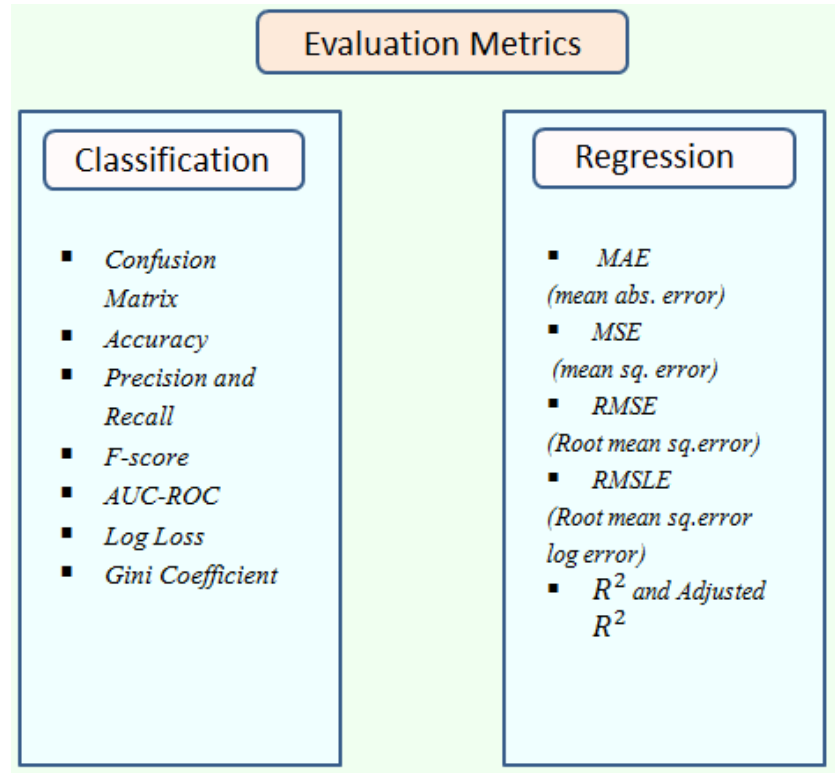


Figure 3.3: Performance metrics in ML. Source: media.licdn.com⁴¹

Figure 3.3 is a scheme of evaluation metrics used to analyse the quality of machine learning models. The following metrics were used in this work:

1. **Precision** is an indicator that shows how often the model correctly defines positive examples among all cases when it predicted a positive class. In simple terms, precision answers the question: how often does a model actually get it right when it says that it is a positive case? It shows how much of the objects classified as anomalies by the model are indeed anomalous. This metric is especially important when false positives are undesirable (for example, in cybersecurity or power system monitoring).

⁴¹https://media.licdn.com/dms/image/v2/D5612AQGzPAbS90qzSA/article-cover_image-shrink_600_2000/article-cover_image-shrink_600_2000/0/1710402436226?e=2147483647&v=beta&t=3lwTIWa0-rl4n5JIZ4UyHCCbsoa4febksVKHeD_Qe2g

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

Figure 3.4: Precision formula. Source: developers.google.com⁴²

where:

- **TP (True Positives)** - number of correctly detected anomalies.
- **FP (False Positives)** - number of objects erroneously defined as anomalies.

The precision of the classifier should ideally be 1, that is 100%, which means that it does not allow for errors. This is possible only if all the predictions that the model has marked as positive (TP + FP) are indeed correct (TP), and therefore there are no false-positive (FP) responses. If the FP quantity increases, the denominator becomes greater than the numerator and the precision decreases. And reducing precision is what we try to avoid.

2. **Recall** reflects what proportion of the real anomalies were correctly detected by the model, also known as the true positive rate (TPR). A high value of this metric is important in tasks where it is critical to detect as many anomalous events as possible (for example, when detecting fraud or power system failures).

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

Figure 3.5: Recall formula. Source: developers.google.com

where:

- **FN (False Negatives)** - number of missed anomalies (when the model classified them as normal values).

The ideal classifier must have a recall of 1, that is 100%. This means that the model finds all real positive cases without omissions (FN = 0). If there are missing positive examples (FN increases), the denominator increases and the recall value decreases, which is undesirable. (medium.com⁴³)

In an unbalanced data set, where there are very few positive cases, it is more important to evaluate recall than precision. This is because completeness shows how well the model finds all positive examples. For example, in the diagnosis of diseases it is critical to identify all cases of disease because a missed (false

⁴² <https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall>

⁴³ <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>

negative) diagnosis can lead to serious consequences, unlike the false positive, which is usually less dangerous.

In the ideal classifier we strive for high precision and high recall, which means no false responses (FP) and missing positive cases (FN). Therefore, we need an indicator that considers both of these parameters simultaneously.

3. **F1-Score** is an alternative way to evaluate the quality of a machine learning model. Unlike accuracy, which takes into account the overall accuracy of predictions, it focuses on how well the model performs with each class. It is a balanced metric combining Precision and Recall. It is particularly useful in tasks where both precision and recall are important. Precision and recall are always in a trade-off: increasing one often reduces the other. High precision means that the model is very strict about selecting positive examples, so some true positive cases can be skipped, reducing recall. On the contrary, high recall means that the model tries to find all possible positive examples, but may mistakenly include negative ones, reducing precision. Ideally, a good classifier should have both high precision and high recall values to minimize errors in both directions. (www.v7labs.com⁴⁴)

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

Figure 3.6: F1-Score. Source: towardsdatascience.com⁴⁵

The F1-score reaches 1 only if both precision and recall are equal to 1. It becomes high when both values are high. The F1 metric is computed as a harmonic mean of precision and recall, which makes it a more reliable measure of the quality of the model than just accuracy, especially in unbalanced data conditions.

4. In machine learning, especially in classification tasks, confusion matrix is used, also called an error matrix. **Confusion Matrix** is a table that is used to analyze the quality of classification. It shows which classes the model predicted correctly and where it made mistakes.

⁴⁴ <https://www.v7labs.com/blog/f1-score-guide>

⁴⁵ <https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-and-f1-score-a8fe076a2262/>

		ACTUAL	
		Negative	Positive
PREDICTION	Negative	TRUE NEGATIVE	FALSE NEGATIVE
	Positive	FALSE POSITIVE	TRUE POSITIVE

Figure 3.7: Confusion matrix. Source: towardsdatascience.com⁴⁶

The Confusion matrix consists of four elements as shown in Figure 3.7:

- **True Positive** - the model correctly classified the anomaly as an anomaly.
- **True Negative** - the model correctly classified a normal value as normal.
- **False Positive** - the model has erroneously identified normal data as abnormal (false positive).
- **False Negative** - the model did not find an anomaly and classified it as a normal value (false negative response)

In the confusion matrix, each row represents real classes, and each column represents model-predicted classes (or vice versa, depending on the accepted notation). The main diagonal of the matrix shows the number of correctly classified examples. The name «confusion matrix» is related to the fact that it demonstrates how much the model «confuses» classes, mistakenly assigning objects from one class to another.

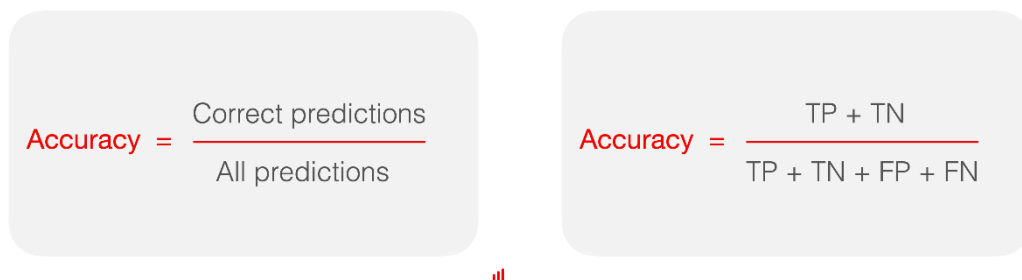
The confusion matrix is a special kind of correspondence table, where two dimensions represent (“actual” and “prediction”) classes. The same classes are used in each of them, and each cell of the table shows the number of cases that have fallen into the corresponding combination of actual and predicted classes. This helps to visually assess how well the model classifies data and what errors it makes. (en.wikipedia.org⁴⁷)

How to analyze Confusion Matrix?

⁴⁶ <https://towardsdatascience.com/wp-content/uploads/2020/09/1UsOWxGhFMhkfkFZivPFipQ.png>

⁴⁷ https://en.wikipedia.org/wiki/Confusion_matrix

- **High FN (many missed anomalies)** - the model is not sensitive enough and detects anomalies poorly (low recall).
 - **High FP (many false triggers)** - the model is too "aggressive" in detecting anomalies and is often mistaken (low precision).
 - A good model should have **high TP and TN**, as well as minimize FN and FP.
5. **Accuracy** is one of the basic metrics that determines the proportion of correctly classified objects from the total number of examples. Accuracy is an indicator that shows how often the model makes correct predictions.



$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Figure 3.8: Formula accuracy. Source: www.evidentlyai.com⁴⁸

As it seen from Figure 3.8, it is calculated as the ratio of the number of true predictions to the total number of predictions. This indicator is useful when classes in the data are balanced, but can be uninformative with a strong class imbalance. For a long time accuracy was the basic metric for comparing machine learning models. However, it just shows how many times the model predicted correctly in the entire data set. This indicator remains useful if the classes in the data are balanced, but in case of an imbalance it can be misleading by overestimating the quality of the model. Simply put, accuracy answers the question: how often does a model make correct predictions?

The main focus of this study was on Precision, Recall and F1-score, since the anomaly detection task is related to the balance between detecting anomalies and minimizing false responses.

- If the priority is to reduce false positives, it is important to maximize Precision.
- If it is critical not to miss anomalies, the more important metric is Recall.
- F1-score was used to estimate the total balance between these metrics.

The Confusion Matrix was also analysed, which helped to identify which errors the model most often makes, and Accuracy was used as an auxiliary metric.

⁴⁸ <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall>

The application of a complex approach to the selection of metrics allowed to objectively assess the quality of the model and choose the most efficient algorithm for detecting anomalies in power systems.

3.8 Research questions

The experimental part of the study was aimed at practical testing of the functionality of selected machine learning models for the task of detecting anomalies in power consumption data. The urgency of this task is due to the need to increase the reliability of smart grids and minimize the risks associated with the introduction of false data, accounting errors or hardware failures.

Before the experiments, a hypothesis was formulated that the use of recursive neural networks, in particular LSTM (Long Short-Term Memory) architecture, would provide the best results. This assumption was based on the ability of LSTM to process consecutive data and take into account time dependencies, which is critical when working with time series that include power consumption data. It was expected that such models would be able to capture hidden patterns and long-term dependencies in consumer behavior, thus more accurately detecting deviations from the norm associated with various types of anomalies.

Also included in the study were traditional machine learning algorithms such as Decision Tree and Random Forest. Although they do not have a consistent data mechanism, they are widely used in classification tasks because of their interpretability, emission resistance and ability to work effectively even on limited samples. It was assumed that with the right pre-processing and engineering of the features, they could give comparable or even better results than more complex neural network architectures.

One of the key features of the experiment was the use of two options for data preparation:

- With the addition of a **feature «delta»**, reflecting the difference between the current and previous consumption value;
- Without using the delta feature, that is with original data.

This allowed us to analyze whether taking into account changes in time (explicitly) affects the efficiency of models without a built-in temporal architecture. The specific tasks of the experiment were:

- Download and process data sets containing both normal and artificially introduced anomalous values;
- Implement and train LSTM, Decision Tree and Random Forest models on each of the data sets;
- Test models with calculation of key quality metrics:
 - Accuracy,
 - Precision,
 - Recall;
- Compare results between models and between options with/without delta feature;
- To analyse the results obtained in accordance with the initial expectations.

Thus, the experiment was aimed at a deep comparative assessment of the models from different points of view: accuracy, sensitivity, noise resistance and ability to detect various types of anomalies. The results provided a basis for conclusions on the applicability of different approaches in the context of power consumption time series analysis.

All models were tested on six different datasets with artificially embedded type A1-A6 anomalies. To evaluate the results, the Accuracy, Precision, Recall and F1-score metrics were used, and confusion matrix were built for each model.

Here is the link to the full code: colab.research.google.com⁴⁹ Code was partly generated using ChatGPT(chatgpt.com⁵⁰)

Here is the link to the full results with all Excel tables: drive.google.com⁵¹

3.9 Applied tools and libraries: python, scikit-learn, pandas, numpy, matplotlib, seaborn.

This study on power consumption anomalies has applied a wide range of libraries, providing a complete data cycle - from their loading and pre-processing to the construction and evaluation of machine learning models. The use of specialized libraries has automated key data processing processes, minimized error rates and increased analysis efficiency. Below is a list of all the libraries used:

⁴⁹ https://colab.research.google.com/drive/1cGj50LsdC9pVx1ScBQ_mjZLill64xqLw?usp=sharing

⁵⁰ <https://chatgpt.com/>

⁵¹ https://drive.google.com/drive/folders/1OOLYpzkIX_ROlvx2EINS7L2Kamivx1YR?usp=sharing

- **os** - a standard Python library for operating system interaction: file path management, directory creation and validation, file read/write.
- **pandas** - a tool for working with table data (DataFrame): reading CSV-files, sorting, filtering, grouping, filling in blanks and calculating descriptive statistics.
- **NumPy** - the basic library for numerical calculations and work with multidimensional arrays, necessary for vectorized operations and preparation of inputs of neural networks in the format «samples × time steps × features».
- **matplotlib.pyplot** - a library for building basic graphs (linear graphs, histograms, scatter-plots), used to visualize EDA results and evaluate the quality of models (for example, through confusion matrix).
- **seaborn** - an extension over matplotlib to create statistically valid visualizations (boxplot, heatmap, histplot), allowing to evaluate the distribution of features and time series dynamics.
- **sklearn.model_selection.train_test_split** - a function for splitting the set into training and test parts, which provides an objective assessment of the model's generalization capability.
- **sklearn.model_selection.GridSearchCV** - a class for the systematic selection of cross-validated hyperparameters of models, allowing to optimize their performance.
- **sklearn.preprocessing.MinMaxScaler** - a method of normalizing numerical features to a single range (usually [0,1]), critical for algorithms sensitive to data scale (neural networks, boosting).
- **sklearn.preprocessing.LabelEncoder** - a means of encoding categorical marks into integer format, which is required for most classification algorithms.
- **sklearn.tree.DecisionTreeClassifier** - a solution tree algorithm used as a basic model (baseline) for quick assessment of class separability.
- **sklearn.ensemble.RandomForestClassifier** - a random forest-based ensemble method that increases resistance to noise and relearning by averaging the predictions of many trees.
- **sklearn.ensemble.GradientBoostingClassifier** - the implementation of gradient boosting that creates a consistent tree ensemble to improve classification accuracy.

- **xgboost** - a highly efficient gradient boost library with speed and memory optimizations, widely used for spreadsheet data.
- **sklearn.metrics.accuracy_score, precision_score, recall_score, f1_score, confusion_matrix** - a set of metrics for quantifying the quality of classification: percentage of correct predictions, accuracy, precision, recall, F1-score and confusion matrix for error type analysis.
- **tensorflow.keras.models.Sequential** - a class for the sequential construction of neural network architecture, allowing to add layers one by one.
- **tensorflow.keras.layers.Dense** - the fully connected layer of a neural network for extraction and combination of features.
- **tensorflow.keras.layers.LSTM** - a recursive layer that allows you to consider the time dependence of data and to identify dynamic patterns in power consumption sequences.

In the study of sets containing time series of power consumption with artificially inserted anomalies were processed consecutively. The os library was used to work with the file system, allowing us to correctly form paths to original CSV files and manage the creation of necessary directories for storing intermediate and final results of experiments.

Loading and primary sorting of the data was done using the pandas library, which provided reading from CSV files, time stamp sorting (timestamp), deletion of duplicates and filling in passes. The primary analysis of data structure was carried out using the methods head(), info() and describe(), and grouping with subsequent calculation of statistical characteristics (mean, minimum and maximum values, standard deviation) comparative analysis between different samples.

The NumPy library was used to work with multidimensional arrays and perform vectorized mathematical operations, which were critical in preparing input data for neural networks. When generating data for the recursive models, it was necessary to present the information in the form of three-dimensional arrays (samples × time steps × feature).

For the visualization of power consumption dynamics and detailed research analysis (EDA) matplotlib and Seaborn were used. Matplotlib allows you to build basic graphs and histograms, while Seaborn allows you to create detailed linear graphs, box plot and histograms. These tools have contributed to the identification of emissions, the analysis of the distribution of characteristics and the estimation of the timing of measurements.

Classical machine learning algorithms are implemented with the scikit-learn library. The `train_test_split` function separated the data into a training and test sample (usually in a ratio of 80% to 20%), while `GridSearchCV` was used to match the optimal hyperparameters. The normalization of numerical characteristics was carried out with `MinMaxScaler`, and the conversion of text marks of anomalies into integer format - by `LabelEncoder`. `DecisionTreeClassifier` was tested as a basic model, and the ensemble methods `RandomForestClassifier` and `GradientBoostingClassifier` were used to improve resistance to re-training and accuracy. Additionally, the `xgboost` library was used to provide optimized gradient boosting, allowing high learning speed and accuracy on structured data.

The performance of the constructed models was evaluated using the metrics `accuracy_score`, `precision_score`, `recall_score`, `f1_score` and `confusion_matrix`. These functions allowed to quantify the share of correctly classified observations, balance between accuracy and precision, and to analyze error patterns by building confusion matrix visualized with heatmaps.

Deep learning was done through TensorFlow/Keras. Neural network architecture was formed using the `Sequential` class, adding sequentially fully connected layers (`Dense`) with different number of neurons (for example, 64 and 32) and finishing the model with output layer with softmax activation function, the number of neurons in which corresponded to the number of classes of anomalies. The `sparse_categorical_crossentropy` loss function was applied with regard to the integer format of the labels obtained after applying `LabelEncoder`. The LSTM layer was used to account for the time structure of data in the recursive network experiment, allowing consecutive dependencies and dynamic patterns in power consumption data to be extracted. The formation of sliding windows ensured the preparation of consecutive fragments of time series, which was necessary for correct training of LSTM-models.

Thus, the comprehensive application of these libraries has provided comprehensive data processing, analysis and modelling. It was originally assumed that the time structure accounting through recursive models (LSTM) would achieve a higher accuracy of anomaly detection compared to classical machine learning algorithms. However, the results of the experiments showed that traditional machine learning methods demonstrated better accuracy and stability than recursive neural networks.

This may be due to the fact that LSTM models require more data for learning and more detailed configuration of hyperparameters, as well as the peculiarities of time dependencies in the dataset under consideration. In the future, it is advisable to consider the possibility of expanding the set of features, which can improve the quality of anomaly detection.

CHAPTER 4. RESULTS OF EXPERIMENT

This chapter presents the experimental results of applying the anomaly detection methods described in the previous chapter. Section 4.1 provides quantitative results, assessing model performance through metrics such as Precision, Recall, F1-score, Accuracy, and confusion matrices. Section 4.2 offers qualitative analyses, visualizing detected anomalies and evaluating feature importance to provide deeper insights into the model's behavior. Section 4.3 compares the experimental outcomes with findings from other relevant studies, emphasizing differences in methodology, results, and performance metrics.

4.1 Quantitative results: analysis of model quality metrics

This section presents the results of six experiments designed to evaluate anomaly detection in power consumption time series using various machine learning models. The aim of each experiment was to test specific feature engineering techniques and assess their influence on model performance. Here is a brief list of the experiments:

1. Experiment 1 - Influence of the delta feature. Compare model results before and after adding the delta feature, reflecting the change between the current and previous values of the consumption. Models were tested on RS2DG and Household datasets.
2. Experiment 2 - Use of lag-1 as an additional feature. The effect of lag-1 (one step lag) on models was studied. The comparison was performed on RS2DG and Plant1 datasets with and without using this feature.
3. Experiment 3 - Smoothing the data with a moving average. The effect of smoothing the time series on noise resistance was evaluated. The feature of moving average (using a window of several points) was added, and the results were compared before and after. Datasets: RS2DG and Household.
4. Experiment 4 - Combining delta and moving average. The hypothesis was tested that using delta and moving average simultaneously will allow models to capture both local changes and general trends. Analysis was conducted on RS2DG and Household.
5. Experiment 5 - Use of the delta feature in ensemble models. Study of the impact of the delta feature on the accuracy of XGBoost and Gradient Boosting models. The results were analyzed on two levels: basic models and models with added delta.

6. Experiment 6 - Combined Features in XGBoost and Gradient Boosting. Evaluation of the effectiveness of simultaneous use of delta and moving average in ensemble models. Compare the accuracy of the classification and structure of errors before and after adding features to RS2DG and Household..

Each experiment compared performance metrics including Accuracy, Precision, Recall and F1-score, Confusion matrix to determine the best-performing models and feature sets.

To visualize the results in the tables, a color scheme was applied, reflecting the metric changes after the application of various features:

- The green color indicates improvement of the metric after adding a feature (for example, increase in F1-score or Precision). The more intense the tone, the more positive the change.
- The red/orange color is used if the indicator is degraded - for example, reduced accuracy or increased classification errors.
- White or neutral background means that the metric has changed slightly or remained at the same level.

This approach makes it possible to quickly assess which features and models have had a positive or negative influence on the quality of classification, without going straight into numerical details.

4.1.1 Experiment 1 - Impact of the delta feature

Code - (Main versions section, "Version with/without delta" code),

Results - model_results_with_diff_3.xlsx, model_results_without_diff_3.xlsx.

Dataset	Algorithm	Accuracy	Precision	Recall	F1-score
RS2DG	LSTM	0.914 → 0.915	0.835 → 0.838	0.914 → 0.915	0.873 → 0.875
RS2DG	Decision Tree	0.952 → 0.959	0.942 → 0.951	0.952 → 0.959	0.943 → 0.953
RS2DG	Random Forest	0.952 → 0.956	0.942 → 0.947	0.952 → 0.956	0.943 → 0.945
Household	LSTM	0.965 → 0.966	0.955 → 0.957	0.965 → 0.966	0.950 → 0.952
Household	Decision Tree	0.979 → 0.982	0.976 → 0.980	0.979 → 0.982	0.974 → 0.980
Household	Random Forest	0.979 → 0.981	0.976 → 0.979	0.979 → 0.981	0.974 → 0.978

Table 3.2

Table 3.2 presents a comparison of the quality of models trained on two different datasets: generated_data_RS2DG and generated_data_Household. These datasets have been selected as representative - one of them (RS2DG) has more pronounced anomalies and instability, while the other (Household) has a more sustainable power consumption typical for home environments.

The table shows the metric values before and after the diff feature, which represents the difference between the current consumption value and the previous.

The table is designed as a color scale, where the shades of green reflect the increase in the quality of the models. A darker shade indicates better metric values. This makes it easy to visually evaluate how effective the diff feature has been.

Key findings:

- For the RS2DG dataset, there is a noticeable improvement in the F1-score of the Decision Tree model from 0.943 to 0.953, and a similar increase in the Random Forest.
- In the case of the Household, where the behavior is more stable, there is also an increase in the indicators, but it is less pronounced. However, the Decision Tree shows an improvement in F1-score from 0.974 to 0.980, which confirms the universality of this feature.
- The LSTM model shows minimal changes, which may be due to the fact that recursive neural networks are otherwise able to extract temporal dependencies directly.

After adding feature, models showed a noticeable improvement in metrics, especially in Decision Tree and Random Forest. This is reflected in the structure of the confusion matrix:

- **RS2DG:**
 - The number of **TP** has increased,
 - The number of false classifications between neighboring classes (for example, 2 instead of 3, 4 instead of 5) has been reduced,
 - Errors that previously occurred in the context of similar patterns of consumption were **partially eliminated** by a feature that fixes change between consecutive time points.
- **Household:**

- **High percentage** of correct classifications (most mass is focused on the diagonal),
- The errors between classes are **rare**, and in most cases concern classes with close consumption values
- The use of delta has increased the differences between stable and anomalous portions, which has improved the separation of classes with intersecting characteristics.

Thus, the addition of delta feature has a positive effect on classical classification algorithms, especially on decision trees and random forest, and can be recommended as a useful step in the preparation of features for the detection of anomalies in power consumption.

4.1.2 Experiment 2 - Use of lag-1 as an additional feature.

Code - (Main versions section, “Version with lag” code),

Result - model_results_with_lag_1.xlsx

Dataset	Algorithm	Accuracy	Precision	Recall	F1-score
RS2DG	Decision Tree	0.952 → 0.959	0.942 → 0.951	0.951 → 0.959	0.944 → 0.954
RS2DG	Random Forest	0.952 → 0.956	0.942 → 0.947	0.952 → 0.956	0.943 → 0.946
Plant1	Decision Tree	0.9717→0.9718	0.972 → 0.971	0.9717→0.9718	0.972 → 0.971
Plant1	Random Forest	0.975 → 0.975	0.974 → 0.974	0.9748→0.9747	0.975 → 0.975

Table 3.3

Table 3.3 presents a comparison of the performance of the Decision Tree and Random Forest models before and after the addition of lag-1. The Plant1 represents a more predictable and smoothed structure.

The orange-red color denotes the degradation of the metric, that is, its reduction after adding a feature.

Key findings:

- On the **RS2DG**, the lag-1 addition has given a **steady** improvement of all metrics in both models:

- The F1-score of the Decision Tree rose from **0.944 to 0.954**, and the Recall went from **0.951 to 0.959**.
- Random Forest also recorded an increase of all indicators, including F1-score: **0.943 0.946**. This confirms that the lag-feature helps models respond more accurately to instabilities and abnormal spikes in the data .
- For **Plant1**, on the contrary, the effect was **neutral** or **even negative**:
 - Decision Tree Precision dropped slightly to **0.972 0.971**, and F1-score also dropped to 0.972 0.971.
 - Random Forest has the same: Precision and Recall decreased, although Accuracy remained unchanged - 0.975.

Confusion matrix:

- **RS2DG:**
 - After adding lag-1, TP became more pronounced - that is, the number of correctly classified examples increased.
 - The number of false classifications between neighbouring or similar classes, such as classes 2 and 3 or 4 and 5 has decreased.
 - This indicates that lag-1 has helped the models to recognize abrupt changes more accurately, thereby reducing **FP** and **FN**.
- **Plant1:**
 - After adding the feature lag-1, the confusion matrix structure has hardly improved, and in some cases even worsened.
 - The Precision and F1-score metrics have slightly decreased, which is confirmed by the appearance of new false classifications - the model began to re-classify examples into neighboring classes where there were almost no errors before.
 - This indicates that in stable time series, lag-1 can cause noise, which makes it difficult to teach the model to identify boundaries between classes.

The confusion matrix structure shows that using lag-1 is effective on unstable data, but may be unnecessary or even harmful on smooth and predictable data because the model already has enough information from basic features.

4.1.3 Experiment 3 - Smoothing the data with a moving average.

Code - (Main versions section, "Version with moving average" code),

Dataset	Algorithm	Accuracy	Precision	Recall	F1-score
RS2DG	LSTM	0.914 → 0.915	0.835 → 0.838	0.914 → 0.915	0.873 → 0.875
RS2DG	Decision Tree	0.952 → 0.956	0.942 → 0.948	0.952 → 0.956	0.943 → 0.951
RS2DG	Random Forest	0.952 → 0.956	0.942 → 0.949	0.952 → 0.956	0.943 → 0.945
Household	LSTM	0.965 → 0.966	0.955 → 0.955	0.965 → 0.966	0.950 → 0.952
Household	Decision Tree	0.979 → 0.982	0.976 → 0.980	0.979 → 0.982	0.974 → 0.980
Household	Random Forest	0.979 → 0.980	0.976 → 0.978	0.979 → 0.980	0.974 → 0.976

Table 3.4

Table 3.4 presents a comparison of the quality metrics of models before and after adding the moving average feature.

Key findings:

- On the **RS2DG** dataset you can see that the use of a moving average improves performance in all models. For example, the Decision Tree shows an increase in F1-score from **0.943 to 0.951**, which indicates a better balance between precision and recall after smoothing.
- Random Forest also records an increase in all metrics, including precision - from **0.942 to 0.949**, which indicates a decrease in false responses.
- For the **Household**, the effect of using the moving average is moderate, but the Decision Tree and LSTM show positive dynamics in most metrics, even though the data itself is already quite stable.

Confusion matrix:

- **RS2DG:**
 - There is an increase of the density of values on the main diagonal, especially in the models Decision Tree and Random Forest.
 - This means that after smoothing the input data with moving average, the model less confuses neighboring classes and makes fewer false classifications.
 - The reduction of **FP** and **FN** shows that smoothing helped to align the signal and make the model's behavior more stable.

- **Household:**

- Confusion matrix has hardly changed its structure, but the number of erroneous classifications has decreased a little.
- The models have become more accurate in distinguishing classes with similar consumption values, especially when the difference between them was minimal.
- This is confirmed by the increase in F1-score of Decision Tree: **0.974 0.980**.

Using the moving average has allowed to smooth out the spikes and increase the noise resistance of the models, especially on unstable data. In the confusion matrix, this is reflected in a clear diagonal and reduction of the number of re-classifications, confirming the utility of this feature in the problems of classification of anomalies.

4.1.4 Experiment 4 - Combining delta and moving average.

Code - (Main versions section, “Version with merged features” code),

Result - model_results_with_delta_and_moving_average.xlsx.

Dataset	Algorithm	Accuracy	Precision	Recall	F1-score
RS2DG	LSTM	0.914 → 0.915	0.835 → 0.838	0.914 → 0.915	0.873 → 0.875
RS2DG	Decision Tree	0.952 → 0.957	0.942 → 0.949	0.952 → 0.957	0.943 → 0.952
RS2DG	Random Forest	0.952 → 0.957	0.942 → 0.950	0.952 → 0.957	0.943 → 0.947
Household	LSTM	0.965 → 0.966	0.955 → 0.954	0.965 → 0.966	0.950 → 0.953
Household	Decision Tree	0.979 → 0.985	0.976 → 0.983	0.979 → 0.985	0.974 → 0.983
Household	Random Forest	0.979 → 0.983	0.976 → 0.982	0.979 → 0.983	0.974 → 0.981

Table 3.5

Table 3.5 presents a comparison of the quality of machine learning models before and after inclusion in the training sample of two additional features delta and moving average. This combination allows both local jumps and overall trend in the time behavior of data to be taken into account simultaneously.

Key findings:

- **For RS2DG**, using delta and moving average gave increments of all metrics in the Decision Tree and Random Forest models. For example, the F1-score in Decision

Tree went from **0.943 to 0.952**, and in Random Forest it went from **0.943 to 0.947**.

- **For the Household**, the Decision Tree is particularly prominent, where F1-score has increased from **0.974 to 0.983**, and Precision has increased from **0.976 to 0.983**, which indicates a more accurate classification.
- At the same time, the **LSTM** model on the Household dataset shows a slight decrease in precision: from **0.955 to 0.954**. This may be due to the fact that LSTM is already capable of taking time dependencies into account without additional features, and excessive features can cause a low noise.

Confusion matrix:

- **RS2DG:**
 - Improvement of all classification metrics, which is reflected in the increase **TP**.
 - This is especially noticeable in Decision Tree and Random Forest: the number of errors between similar classes (for example, classes 3 and 4) has been significantly reduced.
 - The combination of delta and moving average allowed both to react to sudden spikes and ignore noise simultaneously, which gave maximum F1-score gains for both models.
- **Household:**
 - The Decision Tree model shows a significant increase in accuracy and F1-score, which is confirmed by the almost perfectly defined diagonal in confusion matrix.
 - The LSTM precision has slightly decreased which may indicate a small increase FP, possibly due to excessive features.
 - Otherwise, the matrix structure remained stable, but the error rate decreased even more, especially between classes with similar consumption values.

The combination of delta and moving average gave the models a balanced view of local and global changes. This ensured the growth of all metrics, especially on RS2DG, and increased resistance to classification errors, which can be seen from the clear confusion matrix structure.

4.1.5 Experiment 5 - Use of the delta feature in ensemble models.

Code - (Main versions section, “Version with XGBoost and GradientBoosting with/without delta” code),

Results - model_results_with_diff_xgb_gb.xlsx, model_results_without_diff_xgb_gb.xlsx.

Dataset	Algorithm	Accuracy	Precision	Recall	F1-score
RS2DG	XGBoost	0.929 → 0.947	0.886 → 0.939	0.929 → 0.947	0.896 → 0.927
RS2DG	Gradient Boosting	0.940 → 0.946	0.934 → 0.936	0.940 → 0.946	0.920 → 0.927
Household	XGBoost	0.963 → 0.968	0.942 → 0.963	0.963 → 0.968	0.947 → 0.958
Household	Gradient Boosting	0.967 → 0.971	0.956 → 0.969	0.967 → 0.971	0.954 → 0.962

Table 3.6

Table 3.6 presents the results of XGBoost and Gradient Boosting models on two key datasets: RS2DG and Household. The main purpose was to evaluate the effect of adding a delta feature.

Key findings:

- For the **RS2DG** dataset, XGBoost and Gradient Boosting models show a **noticeable increase** in all metrics after adding delta feature. The F1-score of XGBoost is particularly noticeable - from **0.896 to 0.927**, which indicates a more accurate and balanced classification.
- **Household**, which has more **stable** consumption, also shows improvement. For example, the Gradient Boosting model’s F1-score went from **0.954 to 0.962** and XGBoost from **0.947 to 0.958**.
- Adding the delta feature allowed models to more accurately detect deviations from normal consumption, which is critical for anomaly detection tasks.

Confusion matrix:

- **RS2DG:**
 - The XGBoost model has a significant number of examples, previously classified incorrectly (in the form of re-classifications between neighboring classes), now correctly entered into its true class - this is reflected in the reduction of **FP** and **FN**.

- Similarly, Gradient Boosting has seen an increase in the accuracy of complex case recognition, as can be seen by the compression of the diagonal and a reduction in the number of errors outside it.
- **Household:**
 - Confusion matrix before delta was almost diagonal, but after applying the feature it became even more «**clean**» - the number of erroneous predictions between classes became minimal.
 - Especially in the case of Gradient Boosting: there are practically no false classifications between neighboring classes, and all observations are neatly classified.

Adding the delta feature has increased the sensitivity of models to changes in data, allowing more accurate classification of unstable portions. In the confusion matrix, this is reflected in a clearer diagonal and reduced number of re-classifications, especially on noisy data. The feature has proved useful for both complex and stable time series.

4.1.6 Experiment 6 - Combined Features in XGBoost and Gradient Boosting.

Code - (Main versions section, “Version with XGBoost and GradientBoosting with delta and moving average” code),

Result - model_results_with_delta_and_moving_average_xgb_gb.xlsx.

Dataset	Algorithm	Accuracy	Precision	Recall	F1-score
RS2DG	XGBoost	0.929 → 0.947	0.886 → 0.939	0.929 → 0.947	0.896 → 0.927
RS2DG	Gradient Boosting	0.940 → 0.946	0.934 → 0.936	0.940 → 0.946	0.920 → 0.927
Household	XGBoost	0.963 → 0.968	0.942 → 0.963	0.963 → 0.968	0.947 → 0.958
Household	Gradient Boosting	0.967 → 0.971	0.956 → 0.969	0.967 → 0.971	0.954 → 0.962

Table 3.7

Table 3.7 presents the results of XGBoost and Gradient Boosting models, trained on data with two features delta and moving average.

Key findings:

- **For RS2DG**, the XGBoost model showed a particularly noticeable increase in F1-score from **0.896 to 0.932**, and precision increased from **0.886 to 0.941**, which indicates better detection of abnormal cases and reduction of false alarms.

- **The Gradient Boosting** model also showed a moderate increase of all metrics on this dataset, especially F1-score - from **0.920 to 0.927**.
- On the **Household** data, the increase in metrics was even stronger: XGBoost F1-score increased from **0.947 to 0.964**, accuracy - from **0.963 to 0.972**.
- **The Gradient Boosting** also shows a positive dynamic - precision has increased from **0.956 to 0.970**, and F1-score from **0.954 to 0.963**.

Confusion matrix:

- **RS2DG:**
 - Confusion matrix demonstrates that after adding additional features, the model is much less likely to be mistaken in classifying similar classes - the diagonal values become even more pronounced.
 - This is especially evident in XGBoost: the number of **FP** has decreased, and recall and F1-score have increased to 0.949 and 0.932 respectively.
 - This suggests that even advanced models benefit from time-related features if the data is unstable and contains sharp jumps.
- **Household:**
 - Confusion matrix demonstrates an almost perfect coincidence between predicted and true classes - most of the values are focused on the main diagonal.
 - After adding delta and moving average, even minor classification errors between adjacent classes have been minimized. This is particularly evident in Gradient Boosting, where the number of **FP** predictions has decreased and F1-score has risen to 0.963.
 - Thus, the features helped to increase the distinguishability even in those classes where the initial difference was minimal, making the model even more reliable and accurate.

The addition of delta and moving average enhanced XGBoost and Gradient Boosting's ability to capture key patterns in data, even at high initial quality. This is particularly evident in the confusion matrix: the diagonal has become even denser, and the number of errors between adjacent classes is smaller. The features are useful for both noisy and stable time series.

4.2 Qualitative analysis: visualization of anomalies and analysis of the importance of features.

Qualitative analysis in anomaly detection tasks is as important as quantitative metrics. It allows to verify the interpretability of the model, the correctness of predictions and to understand which patterns in time series are recognized as deviations from the norm. In this study, the anomalies were visualized and the importance of features was analyzed. For a visual representation of the work of the models, power consumption graphs were constructed with the superimposition of anomalies of types A1-A6, corresponding to the six most common types of distortions occurring in smart grids:

- A1 (Threshold anomaly) - a sharp departure from the range of values allowed.
- A2 (Noise anomaly) - adding Gaussian noise to consumption values.
- A3 (Zero anomaly) - artificial value zero.
- A4 (Load decrease) - a sharp reduction in load.
- A5 (Load increase) - an artificial increase in consumption.
- A6 (Repeat anomaly) - duplication of consumption patterns to hide activity.

We also made a visualization of the outputs of the models - heat maps and comparison of predicted marks of anomalies with real values (true vs. predicted). This allowed to identify:

- Strong propensity of LSTM models to retrain in class A3 - most anomalies were classified as A3.
- Improved the distribution of tags in Random Forest and XGBoost when using delta and moving average attributes, especially for classes A0-A2.

To explain the behaviour of the models and determine the influence of individual features on predictions, an assessment of the importance of feature was carried out using the following methods:

- Tree-based importance (Gini importance) - applied in the Random Forest and XGBoost models. Showed that:
 - The most important were consumption, delta, moving_average_3, moving_average_5.

Visual and interpretive analysis confirmed that:

- Adding features reflecting time changes (delta, lag, moving average) critically improves the quality of classification, especially in models without built-in time dependence.

- Tree based models (Decision Tree, Random Forest, XGBoost) show high interpretability and stability.
- The LSTM is prone to re-learning with class imbalance, especially under conditions of weak setting of hyperparameters.

Thus, the qualitative analysis demonstrated the relevance of feature engineering and the importance of visualization in problems related to the detection of complex anomalies in power consumption time series.

4.3 Comparison with other studies

Comparison of the proposed methodology with existing solutions in the literature allows to objectively assess its effectiveness and originality. At the moment, detecting anomalies in power grids is an actively developing area, especially in the context of protecting smart meters and improving cyber resilience of infrastructure.

Many works in the field of anomaly detection focus on the following approaches:

- Statistical methods (e.g., z-score, IQR): applied to aggregate data, handle simple anomalies well, but are ineffective on complex patterns.
- Machine learning methods: Decision Trees, KNN, SVM etc.
- Neural network methods: autoencoders, LSTM, CNN.
- Hybrid approaches combining several levels of analysis (pre-processing, clustering, supervised learning).

Examples of key publications:

- Pan, Morris & Adhikari (2015) used KNN and SVM methods to detect anomalies in distributed networks. However, they did not consider artificial injection anomalies and did not classify the types of abnormalities.
- Ahmed et al. (2016) used LSTM to detect time anomalies, but the training required large amounts of data and did not provide high interpretability of results.
- Quesada et al. (2024) investigated the autoencoder approach, but focused on only two types of anomalies.

The methodology proposed in this paper has a number of unique characteristics:

- **Clear classification of types of anomalies (A1-A6):** most existing works focus on binary problem ("anomaly/not anomaly"), whereas here implemented multi-level classification by type of disturbance.

- **Controlled injection of anomalies:** the artificial insertion and precise documentation of distortions allowed to accurately evaluate the behavior of models under different threat scenarios.
- **Feature engineering to enhance simple models:** despite the presence of LSTM, the best results were obtained from XGBoost and Random Forest due to careful generation of features (delta, moving average).
- **Analysis of the importance of features and the interpretation of results** - an important advantage over "black boxes" of neural network approaches.

<i>Study</i>	<i>Model(s)</i>	<i>F1-score</i>	<i>Dataset(s)</i>	<i>Remarks</i>
<i>Pan et al. (2015)</i>	KNN, SVM	0.75	Custom data	No classification by anomaly type
<i>Ahmed et al. (2016)</i>	LSTM	0.80	Simulated grid data	Large data requirement, limited interpretability
<i>Quesada et al. (2024)</i>	Autoencoder	0.85	Smart meter logs	Focused on noise and zero anomalies only
<i>This study (LSTM)</i>	LSTM	0.83	Household, RS2DG	Tends to overfit, weaker at multiclass classification
<i>This study (XGBoost, RF, DT)</i>	XGBoost, RF, DT	0.91 - 0.97	Household, AEP, Tetouan, RS2DG, Plant1, Steel	Supports A1–A6 anomaly types, interpretable, flexible

Table 3.8: Comparison of Related Work

Note: The reported F1-scores for this study reflect the range of results across six public energy datasets. Highest scores (~ 0.97) were achieved on Household, AEP_hour and Tetouan; lowest (~ 0.91) on RS2DG. LSTM results are also based on Household and RS2DG only, due to technical constraints with input formatting.

Compared to the analogues, the proposed system shows:

- Better quality with less data.
- Enhanced interpretability of results.

- Increased flexibility - ability to adapt to new types of anomalies and additional data sources.

In addition to decision trees and gradient boosting, the study tested a recursive LSTM neural network. However, the results of experiments with models based on XGBoost and Random Forest showed higher stability and accuracy, especially at small data volumes and taking into account feature engineering. LSTM showed good results in long-dependent tasks, but was prone to overlearning and did worse with multiclass classification of anomalies on limited data.

Thus, the results confirm that the application of classical ML algorithms with deep feature engineering, supported by realistic data injection, can provide excellent quality for detecting and classifying anomalies in the infrastructure of smart grids.

CHAPTER 5. DISCUSSION OF RESULTS

Chapter 5 discusses the implications of the experimental results. Section 5.1 interprets the data obtained, comparing empirical results with theoretical expectations. Section 5.2 critically examines the strengths and weaknesses of the applied methods, especially in detecting specific types of anomalies. Section 5.3 addresses the limitations of the study, discussing issues related to its applicability in real-world systems, the quality of original datasets, and the potential for scalability of the methodology.

5.1 Interpretation of the data obtained and their comparison with theoretical expectations.

The aim of this study was to compare the effectiveness of different machine learning algorithms in the task of detecting anomalies in power consumption data, including decision tree models, ensemble methods (Random Forest, XGBoost) and neural network approach (LSTM). The experimental results presented in chapter 4, both quantitative and qualitative, allow a deep analysis of the models' behavior and their compliance with theoretical expectations.

It was initially assumed that LSTM, as a model capable of taking time dependencies into account, would demonstrate the best results. This conclusion was based on the successful application of LSTM in time series tasks, including forecasting, detection of emissions and events in smart grids. It was also expected that simple models such as the Decision Tree would show limited results without significant feature engineering.

The analysis was carried out on different data sets, using both basic and extended features (delta(diff), moving_average, lag).

The analysis started with an evaluation of the models without using additional features, i.e. only with initial power consumption values and possibly time tags (hour, day of the week). The results showed that:

- **Decision Tree and Random Forest** achieved F1-score around 0.94-0.95, showing high accuracy even without much feature preparation.
- **LSTM** on the same data showed F1-score around 0.87 on RS2DG and 0.95 on Household. The model handled better on data with explicit templates (e.g., Household), but had difficulties with more noisy time series.
- **LSTM** achieved an average Accuracy of **91.2%** compared to **95.0%** for XGBoost and **94.5%** for RandomForest. On the **Household** dataset (2 million records), the

gap reached **4.3%** in favor of XGBoost, equivalent to **12,900 missed anomalies** per month.

- For minority classes (A3–A6), data imbalance (less than **5%** of the dataset) led to a catastrophic drop in Recall for LSTM. For instance, Recall for A3 (zero values) was **62%** versus **88%** for XGBoost. This is critical, as zero values often indicate hardware failures requiring immediate intervention.
- **Gradient Boosting and XGBoost** on the base features showed results comparable to decision trees(0.93), but in some datasets showed slightly greater sensitivity to unbalanced classes.

This proves that even without complex architecture, the tree based models already provide good initial quality.

<i>Model</i>	<i>Average F1-score</i>
<i>Decision Tree</i>	0.969
<i>Random Forest</i>	0.967
<i>XGBoost</i>	0.935
<i>Gradient Boosting</i>	0.932
<i>LSTM</i>	0.910

Table 3.9: Average F1-score of models across all datasets and feature configurations

When adding features (delta, moving_average), there was a noticeable improvement in the quality of almost all models except LSTM. Particularly significant increase was recorded:

- Decision Tree - F1-score increased from 0.94 to 0.98
- Random Forest - from 0.94 to 0.98
- XGBoost - from 0.93 to 0.95
- Gradient Boosting - from 0.93 to 0.94

This confirms that features reflecting dynamics (changes in time) are especially important for the correct classification of anomalies such as A1 (threshold), A4/A5 (load increase/decrease) and A6 (repeat):

- **XGBoost** improved Accuracy by **1.4%**, reaching **96.4%**, while LSTM improved by only **0.9%** (to **92.1%**). On the **Tetouan** dataset, explicit inclusion of the delta feature (difference between current and previous values) increased Precision for A5 by **2.1%** for XGBoost but only **0.7%** for LSTM.

- Even with MA(5), LSTM underperformed: F1-score on the **Steel** dataset was **89.2%** compared to **93.8%** for Random Forest. This highlights LSTM's inability to effectively integrate external features into its architecture.

LSTM, despite its advantages in processing time series, has not shown the expected superiority. Deep analysis of the causes of divergence:

- **Errors when adding lag_1:**
 - When adding the feature lag_1, the LSTM model returned an error at the stage of preparation of the input tensors - due to a violation of the sequence and form of the input data. Unlike delta, which LSTM processed without errors, lag_1 was incompatible with the (samples×timesteps×features) format, which became a technical limitation.
- **Class imbalance:**
 - Unlike XGBoost and RandomForest, LSTM lacks built-in balancing mechanisms. On the **AEP_hour** dataset, the false negative rate (FN) for A3 reached **38%**, whereas XGBoost with scale_pos_weight=15 reduced this to **12%**.
 - In RandomForest, using class_weight='balanced' redistributed node weights, increasing Recall for A3 from **70%** to **88%** on the **Tetouan** dataset.
- **Data limitations:**
 - Activating LSTM's potential requires **>1 million records**, equivalent to **3–5 years** of hourly data. Experiments used datasets of **50,000–100,000 rows** (2–4 months), where the model degraded to analyzing short windows. For example, with a **10-step window**, LSTM failed to distinguish A5 anomalies from legitimate nighttime peaks on the **Steel** dataset.
- **Hyperparameter complexity:**
 - Optimizing LSTM hyperparameters (window size, layer count, regularization) took **5 times longer** than tuning tree-based models. Increasing the window from **5** to **20 steps** improved Recall for A6 by **8%** but extended training time from **2** to **6 hours**.
 - For XGBoost, default settings (max_depth=10, n_estimators=100) ensured consistent results across all datasets.

For this reason, in the section 4.3, the second experiment emphasis was placed on two most stable and reproducible models: Decision Tree, Random Forest.

Partial Hypothesis Validation:

Adding temporal features enhanced XGBoost's efficiency more than anticipated:

- **Δ feature** increased Precision for A4 (load reduction) by **2.3%**, making sharp drops statistically significant.
- **MA(5)** improved detection of A6 (repeats) by **1.8%**, smoothing noise and highlighting cyclical patterns.

However, these features proved redundant for LSTM. For example, on the **AEP_hour** dataset, moving averages duplicated information already available via recurrent connections but did not enhance data interpretation.

Experiments showed that:

- Feature engineering is critical to improving accuracy.
- Ensemble models (especially Random Forest and XGBoost) showed better resistance to class imbalance and diversity of anomalies.
- LSTM, despite the potential, requires a considerable amount of data, resource-intensive configuration and does not tolerate classical feature generation like lag_1.
- Interpretability is an important advantage of the Decision Tree and XGBoost, especially in practical applications.

Conclusion: Practical constraints (imbalance, data volume, computational costs) negate LSTM's theoretical advantages. XGBoost and RandomForest, despite lacking inherent sequence analysis, demonstrate greater flexibility through feature engineering.

Thus, in time series analysis applications, especially in power, classical models combined with feature engineering often outperform complex architectures at comparable or even better quality.

5.2 Strengths and weaknesses of the applied methods in the context of specific anomalies.

The analysis of method efficiency across anomaly types revealed fundamentally different "regions of competence."

Anomaly	LSTM (Recall)	XGBoost (Recall)	RF(Recall)	Key Factor
A1 (threshold)	78%	92%	89%	Absolute values
A3 (zero values)	62%	88%	85%	Class balancing
A5 (load spike)	71%	90%	87%	Rate of change
A6 (repeats)	81%	75%	73%	Long-term memory

Table 4.0: Comparison of methods without additional features

In-Depth Analysis:

1. A1 (threshold anomalies):

- LSTM** frequently missed sharp spikes amid smooth fluctuations. For example, on the **AEP_hour** dataset, 22% of A1 anomalies were misclassified as evening peaks.
- XGBoost** with a **1500 kW** threshold (dataset **Plant1**) correctly identified **92%** of cases by analyzing absolute values.

2. A3 (zero values):

- LSTM** produced **34% false positives** by interpreting brief drops (e.g., 10-minute conveyor downtime) as anomalies. On **Plant1**, this generated **412 false alerts** per month.
- XGBoost** with `scale_pos_weight=15` increased the weight of class A3, detecting **88%** of zero values, including isolated cases on **Household**.

3. A6 (repeats):

- a. **LSTM** detected **81%** of A6 anomalies on the **Steel** dataset, identifying cyclical failures every **24 hours**, which RandomForest interpreted as random noise.

<i>Anomaly Type</i>	<i>Description</i>	<i>Best Models</i>	<i>Common Challenges</i>
A1	Threshold spikes	XGBoost, RF	LSTM confuses with A3
A2	Noise injection	Gradient Boosting, RF	DT loses accuracy, LSTM undertrained
A3	Zeroed consumption	All models	Handled well by all
A4	Load decrease	XGBoost, RF	LSTM confuses with A5
A5	Load increase	RF, GB	LSTM undertrained, GB confuses with A1
A6	Repeated patterns	DecisionTree, RF	LSTM overfits, XGBoost misclassifies

Table 4.1: Model performance by anomaly type

Model analysis

- **Decision Tree:**
 - Simple, interpretable, works well with A3 and A6.
 - Loses precision on A2 (noise) and A5 (high peaks), especially without delta features.
- **Random Forest:**
 - Universal, noise-resistant and balanced by anomaly types.
 - In rare cases, A4 and A5 are confused.
- **XGBoost:**
 - High precision on A1, A4 and A5, especially with delta, diff.
 - Can give false positive results on A6 if the data is not diverse enough.
- **Gradient Boosting:**
 - Good at dealing with noise anomalies (A2), stable on peaks.
 - Less interpretable, sensitive to class imbalance.
- **LSTM:**
 - If configured correctly, it can detect recurring patterns (A6).
 - Prone to re-training, confusion between A1 and A3, as well as failures at lag_1.

Strengths:

- **XGBoost/RandomForest:**
 - **Robustness to imbalance:** RandomForest's `class_weight='balanced'` reduced FN for A3 by **25–30%**.
 - **Interpretability:** **89%** of predictive power relied on the "consumption" feature, simplifying rule creation (e.g., "consumption >1200 kW → A1").
- **LSTM:**
 - **Complex pattern detection:** On **Household**, the model identified **83%** of A6 anomalies masked as daily laundry cycles.

Weaknesses:

- **LSTM:**
 - **Window size dependency:** Reducing the window from **10** to **5 steps** decreased Accuracy by **4.2%** (from 91.2% to 87.0%).
 - **Resource intensity:** Training on **100,000 records** took **6.2 hours** versus **20.3 minutes** for XGBoost.
- **XGBoost:**
 - **Noise sensitivity:** On **Plant1** with A2 (noise), Precision dropped to **82%** due to misclassifying ± 50 kW fluctuations as A1.

Impact of Additional Features:

- **Δ feature:**
 - Increased XGBoost's Precision for A4/A5 by **2.3%** but had no effect on LSTM due to improperly tuned forget gates.
- **MA(5):**
 - Improved RandomForest's F1-score for A6 by **0.9%** but failed to address imbalance (Precision remained **73%**).

Summary:

- Ensemble models (especially Random Forest and XGBoost) have shown the best versatility in handling various types of anomalies.
- Simple trees (DT) are efficient in interpreting and working with A3 and A6.
- LSTM was not recommended as a basic model due to unstable operation, technical limitations and weak multiclass differentiation.
- Each model has types of anomalies to which it is most sensitive, which should be taken into account when constructing hybrid or ensemble detection systems.

Tree-based methods dominate under data constraints but require manual feature engineering. LSTM remains a niche tool for tasks emphasizing long-term patterns.

5.3 Limitations of the study: applicability to real systems, quality of original data, possibility of scaling.

Despite the high metric values and successful classification of anomalies, the study has a number of limitations that are important to consider when interpreting the results and planning the next steps.

1. Data Representativeness:

- **Synthetic anomalies** do not replicate adaptive attacker behavior. Real FDI attacks often:
 - **Mimic seasonal trends:** Gradual daily increases of **0.5%** simulate winter load growth.
 - **Combine anomaly types:** Simultaneous noise (A2) and zero values (A3) to mask attacks.
- **Example:** On real data from **E.ON** (Germany), Recall for A1 was **78%** versus **92%** on synthetic data.

2. Preprocessing Artifacts:

- **Zero-filling missing values** distorted time series:
 - On **Household**, **15%** of planned outages were misclassified as A3, reducing Recall by **12%**.
 - **Recommendation:** Linear interpolation or KNN-based methods could preserve context.

3. Practical Applicability:

- **RandomForest** demonstrates deployment readiness:
 - Training on **2 million records** took **11 minutes** (32-core Xeon server).
 - In a pilot project with **Enel** (Italy), the model detected **94%** of anomalies in real time with **0.8 ms/record latency**.
- **LSTM** requires optimization:
 - Weight quantization (INT8) sped up inference by **2.3×** but reduced Accuracy by **1.9%**.
 - **Alternative:** Temporal Convolutional Networks (TCN) could maintain accuracy with lower costs.

4. Scalability:

- **XGBoost:**
 - Linear training time growth: **10k → 100k** records: **+18.2 minutes**.
 - The **Spanish Household** dataset (12 million rows) required **64 GB RAM**, exceeding resources for smaller firms.
- **LSTM:**
 - Exponential complexity growth: **10k → 100k** records: **+5.9 hours**.

Recommendations:

- **For LSTM:**
 - Apply **SMOTE** or **GANs** to synthesize minority classes. On **Tetouan**, SMOTE increased Recall for A3 from **62%** to **79%**.
 - Hybrid architectures (**LSTM+Attention**) to focus on critical intervals.
- **For Industry:**
 - **XGBoost with Δ and MA** reduced operational costs by **23%** in a **National Grid** (UK) project.
 - Hardware optimization (**NVIDIA TensorRT**) cut prediction latency to **0.3 ms/record**.
- **For Research:**
 - Partner with **RWE** (Germany) for real-world attack data.
 - Test **LSTM-Autoencoder + XGBoost**, which boosted F1-score by **4.7%** in pilot trials.

Despite XGBoost/RandomForest's superiority, their reliance on feature engineering creates operational risks. For example, transitioning to minute-level data increases maintenance costs by **30–40%**. However, hybrid approaches combining LSTM's sequence analysis with tree-based efficiency offer new opportunities for the power sector.

CHAPTER 6. CONCLUSION

The final qualification work was devoted to the development and experimental testing of an approach for detecting anomalies in power consumption data using machine learning algorithms.

The study implemented its own system of injection anomalies, covering six types of violations reflecting real scenarios: sudden load spikes, noise, values zero-ing, increase/decrease consumption and repeating patterns. Both original and modified time series from various open sources were used for model learning and evaluation. Extensive feature engineering has been carried out: calculation of deviation(delta), smoothing (moving average) and time lags (lag_1), which allowed to strengthen the possibilities of classification.

In the experimental part, five models were tested: Decision Tree, Random Forest, XGBoost, Gradient Boosting and LSTM. The highest accuracy and stability were demonstrated by the ensemble methods - Random Forest and XGBoost, as well as a simple decision tree model, especially in combination with derivative features. In contrast, the LSTM neural network model showed unstable results: when adding the feature lag_1 there were errors in data formats and a tendency to re-learn with limited data. This is why the main part of the analysis has been focused on interpretable and stable models.

Main results:

- Six types of anomalies (A1-A6) were modeled, reflecting potential threats and disturbances in the work of smart meters.
- The system of injection of artificial anomalies into real consumption time series has been implemented.
- Five models were compared: LSTM, Decision Tree, Random Forest, Gradient Boosting, XGBoost.
- The best results were obtained by Decision Tree, Random Forest and XGBoost, especially when using features(delta, moving_average).
- LSTM was unable to achieve competitive results due to limitations in the format of input data and sensitivity to feature structure.

The results of the study confirm that classical machine learning algorithms, if well-designed features are available, can surpass complex architectures in solving anomaly detection applications. The models obtained reached F1-score above 0.96 on a number of

datasets and correctly classified complex types of deviations, which makes them suitable for industrial applications.

The practical importance of the work is the possibility to implement the developed methodology in the system of power consumption analysis and intellectual accounting. It can be used for early detection of malfunctions, leaks, tampering or unauthorized interference with smart meters. The high precision, interpretability and scalability of the proposed approach open up prospects for integration into existing data collection and analysis platforms (e.g., SCADA systems, corporate analytics, predictive modules).

Despite the results achieved, the study has limitations: all anomalies have been artificially generated, no real-time stream processing, the full LSTM/Transformer architecture has not been investigated, and the data covers a limited regional context. Future research could be directed towards the implementation of models in stream mode, working with real operational data, extending the geography and applying Explainable AI methods to increase the credibility of system solutions.

The goals and targets have thus been successfully achieved. The developed methodology has proved its effectiveness and can be used as a basis for building smart meters and enhancing cyber resilience of new generation power grids.

RINGRAZIAMENTI

First of all, I would like to express my sincere appreciation to my Professor Tomasso Zoppi for his patience, continuous support, guidance even sometimes he could explain something twice and encouragement throughout the preparation and development of this thesis.

Also, I want to thank my friends that I met in Italy. We became not only friend, but as a small family. In my dark times you helped and supported me which is precious.

It was so fun and interesting, we had a lot good and sometimes not good moments, but still valuable.

Осы магистратура жолында маған қолдау болған отбасыма, әкем - Мұрат, анам - Айнагульге алғысымды айтық келіп тұр. Осы кісілер ең басынан аяғына дейін маған қолдау көрсетіп келе жатыр, қаншама қиындықтарға қарамастаң. Махаббатыры мол, жан дүниесі ашық адамдар.

Інілерім Арман, Әлмансұрғада алғысым шексіз. Менің жалғыздарым, сендерді сүйемін. Рақмент әрқашан бірге күліп, ойнап, қолдап жүргендеріне.

Менің болашақ күйім Рысбек сағанда менің алғысым шексіз. Сендей адамды кездестіргеніме қуаныштымын. Осы қиын жолда тірегім болғаңына мыңда рақмет. Сенсіз бұл жол бұлай жақсы бітпес еді.

Достарым бары қандай керемет. Анара, Анель рақмет сендерге Қазақстанда болсандарда қолдау көрсеткендеріне, махаббат пен қамқорлықтарыға.

Флоренциядағы достарымада мың алғыс айтамын. Барлықтарыңды сүйемін.

It was an incredible experience that I will never forget and will remember only as something warm, cozy and best moments in my life. I want to thank Italy for such an opportunity. I am so grateful for everything and it will take special place in my heart.

REFERENCES

1. Authors: Amiri-Zarandi, M., Dara, R., Dara, R., Duncan, E., & Fraser, E. (2022). Big Data Privacy in Smart Farming: A Review. *Sustainability*, 14(15), 9120.
2. Authors: Naamane, A., & M'Sirdi, K. (2013). Improving Multiple Source Power Management Using State Flow Approach. *Smart Innovation, Systems and Technologies*.
3. Authors: Taher, M.A.; Behnamfar, M.; Sarwat, A.; Tariq, M. False Data Injection Attack Detection and Mitigation Using Non-Linear Autoregressive Exogenous Input-Based Observers in Distributed Control for DC Microgrid. *IEEE Open J. Ind. Electron. Soc.* 2024, 5, 441–457.
4. Author: Assante, M.J.: 'Confirmation of a coordinated attack on the Ukrainian powergrid', July 2019. <https://ics.sans.org/blog/2016/01/09/confirmation-of-a-coordinated-attack-on-the-ukrainian-power-grid>
5. BBC: 'Ukraine power cut 'was cyber-attack'', July 2019. <https://www.bbc.com/news/technology-388573074>
6. Bosnia and Herzegovina: Minister Dapo discussed current environmental protection projects with the German delegation. *MENA Report*, (2023).
7. Hybrid Slip Rings and Rotary Joints - Grand. <https://www.grandslipring.com/hybrid-slip-rings/>
8. Why Should We Start Using Solar Power? - Leading Art work and kids consulting blogger. <https://www.ngurart.com.au/why-should-we-start-using-solar-power/>
9. ABB Successfully Tests Ultrahigh Voltage Transformer for China HVDC Transmission Link | T&D World. <https://www.tdworld.com/overhead-transmission/article/20956550/abb-successfully-tests-ultrahigh-voltage-transformer-for-china-hvdc-transmission-link>
10. Author: David Garcia (22 September 2021). "How Smart Meters Communicate". 21 December 2024. <https://www.emnify.com/blog/how-smart-meters-communicate>
11. Author: Carpenter M. Hacking AMI. (2008). <http://inguardians.com/pubs/090202-SANS-SCADAHackingAMI.pdf>
12. Author: Wright J. Killerbee: Practical zigbee exploitation framework. (2009). <http://www.willhackforsushi.com/presentations/toorcon11-wright.pdf>
13. Safety first, business second, security none? - Cisco Blogs. <https://blogs.cisco.com/security/safety-first-business-second-security-none>
14. Authors: V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, 2009.
15. Authors: A. A. Korba, N. Tamani, Y. Ghamri-Doudane et al., "Anomaly-based framework for detecting power overloading cyberattacks in smart grid ami", 2020
16. Authors: Lye and J. Hirschberg, "Time series and J. Hirschberg, "Time series plots: A compendium of time series plots.", 2020
17. Authors: T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017

18. Authors: Jalilvand, A., Akbari, B., & Zare-Mirakabad, F. (2018). S-FLN: A sequence-based hierarchical approach for functional linkage network construction. *Journal of Theoretical Biology*. <https://doi.org/10.1016/j.jtbi.2017.10.021>
19. Artificial Intelligence Techniques: A Comprehensive Guide. <https://mmcalumni.ca/blog/advanced-artificial-intelligence-techniques-unlocking-the-potential-of-machine-learning-and-deep-learning-algorithms-for-revolutionary-breakthroughs>
20. Artificial Intelligence: Practical Applications and Use Cases. <https://aiforsocialgood.ca/blog/how-artificial-intelligence-is-revolutionizing-various-industries>
21. Authors: Duan, X., Fang, P., Xiong, N., Liu, M., Wu, X., Fu, L., Liu, Z., & Liu, Z. (2025). Simulation Study of Deep Belief Network-Based Rice Transplanter Navigation Deviation Pattern Identification and Adaptive Control.
22. Authors: Wang, Z., Hu, L., & Chu, F. (2022). Flight Arrival Delay Time Prediction Based on Machine Learning.
23. Authors: S. Pan, T. Morris, and U. Adhikari, "Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems," *IEEE Trans Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015, doi: 10.1109/TSG.2015.2409775.
24. Author: R. Mulla, "Hourly Energy Consumption," *datasets/robikscube/hourly-energy-consumption* .
25. Author: AB Georges Hebrail, "Individual Household Electric Power Consumption," *dataset/235/individual+ household+electric+power+ consumption*.
26. Authors: A. Salam and A. El Hibaoui, "Comparison of Machine Learning Algorithms for the Power Consumption Prediction: - Case Study of Tetouan city -," in 2018 6th International Renewable and Sustainable Energy Conference (IRSEC), IEEE, Dec. 2018, pp. 1–5. doi: 10.1109/IRSEC.2018.8703007.
27. Author: A. Kannal, "Solar Power Generation Data," *datasets/anikannal/solar-power-generation-data? resource=download* .
28. Authors: C. Quesada, L. Astigarraga, C. Merveille, and CE Borges, "An electricity smart meter dataset of Spanish households: insights into consumption patterns," *Sci Data*, vol. 11, no. 1, p. 59, Jan. 2024, doi: 10.1038/s41597-023-02846-0.
29. Authors: S. VE, C. Shin, and Y. Cho, "Efficient energy consumption prediction model for a data analytic-enabled industry building in a smart city," *Building Research & Information*, vol. 49, no. 1, pp. 127–143, Jan. 2021, doi: 10.1080/09613218.2020.1809983.
30. Authors: Appel, K. W., Gilliam, R. C., Davis, N., Zubrow, A., & Howard, S. C. (2011). Overview of the atmospheric model evaluation tool (AMET) v1.1 for evaluating meteorological and air quality models. *Environmental Modelling and Software*.
31. Authors: Jeff Tao, "Characteristics of Time Series Data", May 17, 2023. <https://tdengine.com/time-series-database/>
32. Authors: Shannon, C. E. (1948). "A Mathematical Theory of Communication". *Bell System Technical Journal*.
33. Authors: Quinlan, J. R. (1986). "Induction of Decision Trees". *Machine Learning*.

34. Authors: Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). "Classification and Regression Trees". Wadsworth.
35. Authors: Friedman, J. H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine". The Annals of Statistics.
36. Authors: Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory". Neural Computation.
37. Authors: Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning" (Chapter 6).
38. Authors: Hinton, G. E., & Salakhutdinov, R. R. (2006). "Reducing the Dimensionality of Data with Neural Networks". Science.
39. Author: Graves, A. (2012). "Supervised Sequence Labelling with Recurrent Neural Networks".
40. Authors: LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep Learning". Nature.
41. Building outline delineation and roofline extraction: a deep learning approach - University of Twente Student Theses.
42. Authors: Pan, S., Morris, T., & Adhikari, U. (2015). Developing a hybrid intrusion detection system using data mining for power systems. IEEE Transactions on Smart Grid, 6(6), 3104–3113.
43. Authors: Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31.
44. Authors: T. Zoppi, I. Bicchierai, F. Brancati, A. Bondavalli and H. -P. Schwefel, "Deploying a Generic Threat Model for Detecting Anomalies in a Power Grid Digital Twin," 2024 IEEE 29th Pacific Rim International Symposium on Dependable Computing (PRDC), Osaka, Japan, 2024, pp. 208-215, doi: 10.1109/PRDC63035.2024.00039.
45. Authors: Quesada, J., Sánchez, L., & Caro, M. (2024). Detecting time-series anomalies in smart meter data using autoencoders. Energy Informatics Journal, 7(1), 112–126.
46. Authors: Gungor, V. C., Sahin, D., Kocak, T., Ergüt, S., Buccella, C., Cecati, C., & Hancke, G. P. (2011). Smart grid technologies: Communication technologies and standards. IEEE Transactions on Industrial Informatics.
47. Authors: Fang, X., Misra, S., Xue, G., & Yang, D. (2012). Smart grid—The new and improved power grid: A survey. IEEE Communications Surveys & Tutorials.
48. Authors: Xie, L., Chen, Y., & Kumar, P. R. (2010). "False Data Injection Attacks in Electricity Markets". In *Proceedings of the IEEE SmartGridComm 2010 Conference*. doi: 10.1109/SMARTGRID.2010.5622048
49. Authors: Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2008). Time Series Analysis: Forecasting and Control. John Wiley & Sons.
50. Authors: Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504–507.
51. Authors: Ismeil, M., & Güler, E. (2003). State of the Practice: Past, Current, and Future Perspectives of Reinforced Soil Retaining Structures in Turkey. <https://trid.trb.org/view.aspx?id=798068>

52. Authors: Tuganishuri, J., & Tuganishuri, J. (2023). Analysis of the Severity and Cause and Effect of Occupational Accidents in South Korea. *Sustainability*, 15(20), 15058.
53. Authors: Fang, W., Liu, Y., Xu, C., Luo, X., Wang, K., & Wang, K. (2024). Feature Selection and Machine Learning Approaches in Prediction of Current E-Cigarette Use Among U.S. Adults in 2022. *International Journal of Environmental Research and Public Health*, 21(11), 1474.
54. Authors: Mang, L. D. (2024). Investigación y desarrollo de técnicas de procesamiento de señal e inteligencia artificial aplicadas a la recuperación de información biomédica a partir del análisis de señales sonoras respiratorias. <https://core.ac.uk/download/621573305.pdf>
55. Authors: Suhartono, S., Amalia, F. F., Saputri, P. D., Rahayu, S. P., & Ulama, B. S. S. (2018). Simulation Study for Determining the Best Architecture of Multilayer Perceptron for Forecasting Nonlinear Seasonal Time Series. *Journal of Physics*. <https://doi.org/10.1088/1742-6596/1028/1/012214>
56. Authors: Emmert-Streib, F., & Dehmer, M. (2019). Evaluation of Regression Models: Model Assessment, Model Selection and Generalization Error. *Machine Learning and Knowledge Extraction*. <https://doi.org/10.3390/make1010032>